PB87-127759

Integrated Air Pollution Control System Design and Cost-Estimating Model Version 2 (IAPCS2). User's Manual. Volume 2. Appendix C

PEI Associates, Inc., Cincinnati, OH

Prepared for

Environmental Protection Agency
Research Triangle Park, NC

Sep 86

U.S. Department of Commerce
National Technical Information Service
NTIS

PB87-127759

EPA/600/8-86/031b
September 1986

USER'S MANUAL FOR THE INTEGRATED AIR POLLUTION CONTROL SYSTEM DESIGN AND COST-ESTIMATING MODEL VERSION II

VOLUME II

by

P. J. Palmisano and B. A. Laseke
PEI Associates, Inc.
11499 Chester Road
P.O. Box 46100
Cincinnati, Ohio 45246

EPA Contract No. 68-02-3995
Work Assignment 4

EPA Project Officer

Norman Kaplan

U.S. ENVIRONMENTAL PROTECTION AGENCY
AIR AND ENERGY ENGINEERING RESEARCH LABORATORY
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27711

AIR AND ENERGY ENGINEERING RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
RESEARCH TRIANGLE PARK, NC 27711

REPRODUCED BY
U.S. DEPARTMENT OF COMMERCE
NATIONAL TECHNICAL
INFORMATION SERVICE
SPRINGFIELD, VA. 22161

TECHNICAL REPORT DATA
(Please read Instructions on the reverse before completing)

1. REPORT NO. EPA/600/8-86/031b	2.
3. RECIPIENT'S ACCESSION NO. PB87-127759	
4. TITLE AND SUBTITLE: User's Manual for the Integrated Air Pollution Control System Design and Cost-estimating Model (Version II); Volume II (Appendix C)	5. REPORT DATE: September 1986
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S): P. J. Palmisano and B. A. Laseke	8. PERFORMING ORGANIZATION REPORT NO. PN 3650-4
9. PERFORMING ORGANIZATION NAME AND ADDRESS: PEI Associates, Inc. P.O. Box 46100 Cincinnati, Ohio 45246	10. PROGRAM ELEMENT NO.
	11. CONTRACT/GRANT NO. 68-02-3995, Task 4
12. SPONSORING AGENCY NAME AND ADDRESS: EPA, Office of Research and Development Air and Energy Engineering Research Laboratory Research Triangle Park, NC 27711	13. TYPE OF REPORT AND PERIOD COVERED: User's Manual; 10/84 - 5/86
	14. SPONSORING AGENCY CODE: EPA/600/13

15. SUPPLEMENTARY NOTES: AEERL project officer is Norman Kaplan, Mail Drop 63, 919/541-2556. Volume I is the basic manual and Appendices A and B. EPA-600/8-86-031c is the related disk. PB87-127775

16. ABSTRACT: The manual describes and is a guide to the user of Version II of the Integrated Air Pollution Control System (IAPCS-II), a computerized simulation model for estimating the costs and predicting the performance of sulfur dioxide, nitrogen oxides, and particulate matter control systems for coal-fired utility boilers. It gives the design bases of the modules comprising the model and the structure of the program itself, as well as the bases for a number of model enhancements available to the user. The model includes conventional and emerging technologies that effect pre-, in situ, and post-combustion emission control. The model can accept any combination of the technology modules built into the system. Interactions are reflected in a material balance tabulation of the exit of each module. Alterations in the material balance are used to account for integrated performance and cost effects. The emission control technologies contained in IAPCS-II can be selected in either isolated or integrated configurations. IAPCS-II incorporates a number of enhancements to the design premises of the emission control modules, as well as the model's user access and versatility. Enhancements to the control modules involved upgrades to five modules: wet desulfurization, low-NOx combustion, limestone injection multistage burner (LIMB), electrostatic precipitator, and fabric filter.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS		b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group	
Pollution	Coal	Pollution Control	13B	21D
Cost Estimating	Combustion	Stationary Sources	05A, 04A	21B
Mathematical Models	Emission	Particulate	12A	14G
Economics	Sulfur Dioxide	LIMB	05C	07B
Boilers	Nitrogen Oxides	Fabric Filters	13A	
Utilities	Particles			

18. DISTRIBUTION STATEMENT: Release to Public	19. SECURITY CLASS (This Report): Unclassified	21. NO. OF PAGES: 337
	20. SECURITY CLASS (This page): Unclassified	22. PRICE: $30.95

EPA Form 2220-1 (9-73)

NOTICE

This document has been reviewed in accordance with U.S. Environmental Protection Agency policy and approved for publication. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

ABSTRACT

The Integrated Air Pollution Control System (AIPCS) is a computerized simulation model developed for EPA's Air and Energy Engineering Research Laboratory (AEERL) to estimate the costs and predict the performance of sulfur dioxide (SO_2), nitrogen oxides (NO_x), and particulate matter (PM) emission control systems for coal-fired utility boilers. The model includes conventional and emerging technologies that effect pre-, in situ, and post-combustion emission control. The model can accept any combination of the technology "modules" built into the system. Interactions are reflected in a material balance tabulation of the exit of each module. Alterations in the material balance are used to account for integrated performance and cost effects. The emission control technologies contained in IAPCS can be selected in either "isolated" or "integrated" configurations.

This version of IAPCS (IAPCS-II) was completed in April 1986. It incorporates a number of enhancements to the design premises of the emission control modules as well as the model's user access and versatility. Enhancements to the control modules involved upgrades to the wet flue gas desulfurization (FGD) module, upgrades to the low-NO_x combustion module, upgrades to the limestone injection multistage burner (LIMB) module, and upgrades to the electrostatic precipitator (ESP) and fabric filter (FF) modules. Other important enhancements to IAPCS-II include expanding the solid waste handling and disposal module, housing the model on a microcomputer (personal computer), providing EPRI and TVA economic premises, and expanding the user-activated parameter file.

The User's Manual describes the second version of IAPCS. This manual provides a guide to the user of the model. It presents the design bases of the individual modules comprised by the model and the structure of the program itself, as well as the bases for a number of model enhancements now available to the user.

Since program "bugs" and other errors may be discovered by model users, it is requested that the errors be conveyed to the AEERL project officer by mail (U.S. EPA, MD-4, Research Triangle Park, NC 27711) or by phone (919/541-2556). If and when the model is upgraded, the compiled version (diskette) will be changed and dated to identify it. Users may contact the AEERL Technical Information Service (phone 919/541-2218) to determine the latest version of the model, and how to obtain it.

CONTENTS

Routine	Page (C-)
Uncontrolled Emissions	2
Low-NO_x Burners/Overfire Air Ports	5
Spray Humidification	9
Cost Model	21
Physical Coal Cleaning	50
Dry Sorbent Injection	52
Lime Spray Drying	56
Fabric Filter	68
Electrostatic Precipitator	84
TVA Flue Gas Desulfurization	244
Limestone Injection Multistage Burner (LIMB)	279

Preceding page blank

APPENDIX C

IAPCS SOURCE LISTING

```
$DEBUG
C
      SUBROUTINE UNCNTL
C
C         THIS SUBROUTINE DOES INITIAL SYSTEM PERFORMANCE CALCULATIONS,
C     BOTTOM ASH CALCULATIONS,  AND CALCULATES UNCONTROLLED EMISSIONS.
C     MEMBERS OF COMMON /STREAM/ ARE INITIALIZED HERE.
C
C         WRITTEN BY PEI ASSOC., INC. (JP)
C               5-10-83
C
      INTEGER FMB, CT, BOTTOM, TYPCT, CLNLVL
      LOGICAL KO(2), TVA
      REAL INDCST
      COMMON /COSTS/ CAPCST(10,8), ANN(25), INDCST(10)
      COMMON /COAL1/ HHV, PS, PA, CST, PNA, ALK, CL, FEO, PM
      COMMON /STREAM/ PPH, ACFM, TEMP, PPHMST, PPHALK, PPHPRT, PPHSO2, PPHNOX
      COMMON/INPUT/BSIZE, CF, AHFX, AHXT, HHVROM, PSROM, PAROM, CSTROM, PNAROM,
     &ALKROM, CLROM, FEOROM, PMROM, VROM, FCROM, HHVCLN, PSCLN, PACLN, CSTCLN,
     &PNACLN, ALKCLN, CLCLN, FEOCLN, PMCLN, VCLN, FCCLN, FMB, NR, IS, CT, TYPCT,
     &CLNLVL, BOTTOM, NM, IAPCS(9), KO, TVA
      COMMON /INSYSP/ CCEFF, UTEFF, BNHR, FANLOS, QIN, ALKIN
      COMMON /PARM/ P(500)
C
C     DEBUG OUTPUT VARIABLES
C
C
C     DEBUG OUTPUT
C
C
C     CALCULATE HEAT INPUT FROM PRE-CONTROL SYSTEM PERFORMANCE
C
C>>>> UNIT THERMAL EFFICIENCY
      UTEFF=.257*HHV**.13
C      UTEFF= P(1)-.44362*PM*PM*.0001
C>>>> FAN LOSS (BTU/KWH)
      AUXLOS=19892*HHV**(-.336)
C      FANLOS=.15*3500.*BSIZE*.001
C>>>> BOILER NET HEAT RATE (BTU/KWH)
      IF(P(3) .EQ. 0.0) THEN
         BNHR= P(2)/UTEFF+AUXLOS
      ELSE
         BNHR= P(3)
      ENDIF
C
C>>>> INPUT HEAT RATE (MM BTU/HR)
      QIN= BNHR*BSIZE*.001
C
C     BOTTOM ASH CALCULATIONS
C
C>>>> COMPUTE BOTTOM ASH WASTE (TONS/YR)
C >>> SET BOTTOM ASH FACTOR
```

```
C       BAF= 20.*(1-P(5+BOTTOM))
C       IF(TYPCT.EQ.6.OR.CT.EQ.3) BAF=20.*(1.-P(8))
C       IF (P(4).GT.0) BAF=20.*(1.-P(4))
C       TONASH= BAF*PA*500./HHV*QIN*8760.*CF/2000.
C>>>   ANNUAL BOTTOM ASH QUANTITY (TONS)
C       ANN(13)= TONASH
C
C      FLYASH CALCULATIONS
C
       IF(TYPCT .EQ. 0) GO TO 10
C>>>> TYPES OF TYPICAL COALS
       CT= TYPCT-3
       IF(CT .LE. 4) CT= 1
   10 CONTINUE
C>>>> INITIAL PARTICULATE FLOW (PPH)
C >>> SET EMISSION FACTOR
       EF= P(5+BOTTOM)
       IF(CT .EQ. 3) EF= P(8)
       IF (P(4).GT.0) EF=P(4)
       PPHPRT= 20*EF*PA*500./HHV*QIN
C>>>> INITIAL SO2 FLOW(PPH)
C >>> SET EMISSION FACTOR
       EF= P(8+CT)
       IF (P(5).GT.0) EF=P(5)
       P(5)= EF
       PPHSO2= 40*EF*PS*500./HHV*QIN
C>>>> INITITAL NO2 FLOW (PPH)
C >>> SET EMISSION FACTOR
       EF= P(2*(FMB-1)+CT/3+12)
       IF(BOTTOM .EQ. 2) EF= P(16)
       PPHNOX= 40*EF*500./HHV*QIN
C
C      REMAINDER OF MATERIAL BALANCE CALCULATIONS
C
C>>>> INITIAL FLUE GAS FLOW (ACFM)
C>>>   DEFAULT
       IF(AHFX .EQ. 0) ACFM= QIN*P(17)*(1.+P(18))/60.*(AHXT+460.)/530.
C>>>   USER INPUT
       IF(AHFX .NE. 0) ACFM= AHFX
C>>>> INITIAL FLUE GAS FLOW (PPH) (WET)
       PPH = (1.E4*(100.-PA-P(21))/HHV+P(22)*(785*P(18)+783))*QIN+PPHPRT
C>>>> INITIAL TEMPERATURE (DEG.F)
       TEMP= AHXT
C>>>> INITIAL MOISTURE FLOW (PPH)
C >>> WATER F FUEL (#/MMBTU)
       WF= PM*1.E4/HHV
C >>> WATER F AMBIENT AIR (#/MMBTU)
       WA= (785*P(18)+783)*.013
C >>> WATER F COMBUSTION (#/MMBTU)
       WC= 9.E4*5.5*(1.-(PA+PM)*.01)/HHV
C >>> TOTAL WATER
      PPHMST= QIN*(WF+WA+WC)
```

```
C>>>> INITIA ALKALINITY FLOW (PPH)
      PPHALK= ALK*.01*PPHPRT
      ALKIN= PPHALK
C
C     DEBUG OUTPUT
C
C
      RETURN
      END
```

```
C
C           INTEGRATED AIR POLLUTION CONTROL SYSTEM PROGRAM
C
C
C              LOW NOX BURNERS/OVERFIRE AIR PORTS ROUTINE
C              --- --- ---------------- --- ----- -------
C
C
C     WRITTEN BY:  J. R. SCHOETTELKOTTE
C                  PEDCO COMPUTER SERVICES
C                  MAY, 1983
C
C     WRITTEN FOR:  G. E. BAKER
C                   PEI ASSOC., INC.
C                   PEI PN 3568-3
C                   J. O. MILLIKEN
C                   U.S. EPA; IERL-RTP
C
C
C           THIS ROUTINE CALCULATES DESIGN PARAMETERS, CAPITAL AND
C     ANNUAL OPERATING COSTS, AND MATERIAL BALANCE FOR LOW NOX BUR-
C     NERS. THE ALGORTHIM WAS EXCERPTED FROM  THE EPA LIMBCOST MODEL
C     (LACHAPELLE, 3/86).
C
      SUBROUTINE LNBOF
      LOGICAL    KO, TVA
      INTEGER    BOTTOM, CLNLVL, CT, FMB, TYPCT
      REAL       IC, INDCST
      DIMENSION AC(13), ANN(25), CAPCST(10,8), IAPCS(9), IC(10), INDCST(10),
     &          KO(2), P(500), TABLE(8,8), FLNB(2,2)
      COMMON /COAL1/    HHV, PS, PA, CST, PNA, ALK, CL, FEO, PM
      COMMON /ESC/ IYC, RATPLT, RATLA2, RATMA2, RATOM
      COMMON /COSTS/   CAPCST, ANN, INDCST
      COMMON /FAN/     DELTAP, CSTFAN
      COMMON /INPUT/    BSIZE, CF, AHFX, AHXT, HHVROM, PSROM, PAROM, CSTROM,
     &                  PNAROM, ALKROM, CLROM, FEOROM, PMROM, HHVCLN, PSCLN,
     &                  PACLM, CSTCLN, PNACLN, ALKCLN, CLCLN, FEOCLN, PMCLN,
     &                  FMB, NR, IS, CT, TYPCT, CLNLVL, BOTTOM, NM, IAPCS, KO,
     &TVA
      COMMON /INSYSP/ CCEFF, UTEFF, BNHR, FANLOS, QIN, ALKIN
      COMMON /MATBAL/ TABLE
      COMMON /OPT/     BYPFGD, BYPFF, EFFESP, EFFLSD, EFFLIM
      COMMON /PARM/    P
      COMMON /STREAM/ PPH, ACFM, TEMP, PPHMST, PPHALK, PPHPRT, PPHSO2, PPHNOX
      DATA FLNB/3.7, 1.09, 5.65, 1.37/
C
C
      CSTF= FLNB(FMB, NR)*.966
      IODB = 8
      IO6 = 6
```

```
      IO7 = 7
      DO 10 I=1,13
      AC(I) = 0.
   10 CONTINUE
      DO 20 I=1,8
      IC(I) = 0.
   20 CONTINUE
C
C      WRITE (IODB,8000)
C 8000 FORMAT (T2,'ENTERING SUBROUTINE LNBOF.')
C      WRITE (IODB,8010) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,
C     &                  PPHNOX,QIN,BSIZE,IS,CAPCST(2,1),INDCST,
C     &                  ANN,(P(I),I=61,70)
C 8010 FORMAT (T10,'PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT=',/
C     &          6E12.4,
C     &        /' PPHSO2,PPHNOX,QIN,BSIZE,IS=',4E12.4,I5,
C     &        /T10,'CAPCST(LNBOF)=',F14.4,
C     &        /' INDCST=',
C     &        /8F9.2,
C     &        /' ANN=',7F10.2,
C     &        /T14,6F10.2,
C     &        /T14,6F10.2,
C     &        /T14,6F10.2,
C     &        /T10,'P(61-70)=',/5F10.2,/5F10.2)
C
C*******************************************************************************
C
C     PART 2:  LOW NOX BURNERS.
C
C     THIS PART PERFORMS ALL NECESSARY CALCULATIONS FOR LOW NOX
C     BURNERS.
C
C     DETERMINE NUMBER OF BURNERS REQUIRED.  THIS IS THE ONLY SIZING
C     CALCULATION.
C
      IF (KO(1)) WRITE (IO6,6210)
      IF (KO(2)) WRITE (IO7,6210)
 6210 FORMAT (///T5,'LOW NOX BURNERS')
C
C     CALCULATE DIRECT COSTS.
C
      CAPCST(2,1) = (500./BSIZE)**.6*CSTF*BSIZE*1000.
      CAPCST(2,1) =CAPCST(2,1) * RATPLT
C
C
C      CALCULATE ENGINEERING DESIGN AND SUPERVISION(TVA) OR
C      GENERAL FACILITIES(EPRI)
C
      IC(1) = P(62)/100. * CAPCST(2,1)
      INDCST(1) = INDCST(1) + IC(1)
C
C      CALCULATE A-E CONTRACTOR(TVA) OR ENG. AND HOME OFF. FEES (EPRI)
```

```
C
      IC(2) = P(61)/100. * CAPCST(2,1)
      INDCST(2) = INDCST(2) + IC(2)
C
C     CALCULATE CONSTRUCTION EXPENSE (TVA) PROJECT CONTINGENCY (EPRI)
C
      IC(3) = P(63)/100. * CAPCST(2,1)
      INDCST(3) = INDCST(3) + IC(3)
C
C     CALCULATE CONTRACTOR FEES (TVA) OR PROCESS CONTINGENCY (EPRI)
C
      IC(4) = P(64)/100. * CAPCST(2,1)
      INDCST(4) = INDCST(4) + IC(4)
C
C     SUM DIRECT AND INDIRECT CAPITAL COSTS TO HAVE A BASE FOR THE
C     REMAINDER OF THE INDIRECT COSTS.
C
      COST = CAPCST(2,1) + IC(1) + IC(2) + IC(3) + IC(4)
C
C     CALCULATE CONTINGENCY (TVA) OR SALES TAX (EPRI)
C
      IF (TVA) THEN
            IC(5) = P(67)/100. * COST
         ELSE
            IC(5) = P(67)/100. * CAPCST(2,1)
         ENDIF
      INDCST(5) = INDCST(5) + IC(5)
C
C     INCLUDE IC(5) IN RUNNING COST BASE TOTAL
C
      COST = COST + IC(5)
C
C     CALCULATE ROYALTIES (TVA) OR ROYALTY ALLOWANCE (EPRI)
C
      IC(6) = P(66)/100. * COST
      INDCST(6) = INDCST(6) + IC(6)
C
C     CALCULATE INTEREST DURING CONSTRUCTION (TVA)
C
      IF (TVA) THEN
        IC(8) = P(69)/100. * COST
        INDCST(8) = INDCST(8) + IC(8)
      ENDIF
C
C       CALCULATE ALLOWANCE FOR STARTUP AND MODIFICATIONS (TVA) OR
C                 INITIAL CATALYST (EPRI)
C
      IC(9) = P(65)/100. * COST
      INDCST(9) = INDCST(9) + IC(9)
C
C       CALCULATE LAND (TVA AND EPRI)
C
```

```
      IC(10) = P(181)
      INDCST(10) = INDCST(10) + IC(10)
C
C     MAINTENANCE LABOR AND MATERIALS IS THE ONLY ANNUAL COST
C
      ANN(10)= ANN(10)+P(68)*.01*CAPCST(2,1)
C
C     CALCULATE MATERIAL BALANCE.  ONLY THE FLUE GAS FLOW IN LB/HR
C     AND FLUE GAS NOX ARE CHANGED.
C
      IF(FMB .EQ. 1) THEN
        RNX= .5*PPHNOX
      ELSE
        RNX= .25*PPHNOX
      ENDIF
      PPH= PPH- RNX
      PPHNOX= PPHNOX-RNX
C
C     DEBUG OUTPUT.
C
C      WRITE (IODB,8200) NOBURN,CAPCST(2,1),IC,INDCST,PPH,ACFM,TEMP,
C     &                  PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
C 8200 FORMAT (T10,'NOBURN,CAPCST(LNBOF)=',2F14.4,
C C     &        /T10,'IC=',8F10.2,
C     &       /T10,'INDCST=',8F10.2,
C     &       /T10,'PPH,ACFM,TEMP,PPHMST=',4F14.4,
C     &       /T10,'PPHALK,PPHPRT,PPHSO2,PPHNOX=',4F14.4)
C
C     SKIP TO END OF ROUTINE.
C
      RETURN
      END
```

```
C
C                INTEGRATED AIR POLLUTION CONTROL SYSTEM PROGRAM
C                ========== === ========= ======= ====== =======
C
C                        SPRAY HUMIDIFICATION ROUTINE
C                        ----- -------------- -------
C
C
C
C     WRITTEN BY;  J. R. SCHOETTELKOTTE
C                  PEDCO COMPUTER SERVICES
C                  JUNE, 1983
C
C     WRITTEN FOR:  G. E. BAKER
C                   PEI ASSOC., INC.
C                   PEI PN 3568-3
C                   J. O. MILLIKEN
C                   U.S. EPA: IERL-RTP
C
C
C          THIS ROUTINE CALCULATES DESIGN PARAMETERS, CAPITAL AND
C     ANNUAL OPERATING COSTS, AND MATERIAL BALANCE FOR SPRAY HUMIDI-
C     FICATION.  THE ALGORITHMS WERE DEVELOPED BY B. A. LASEKE AND M.
C     D. GIORDANO AND REVIEWED BY G. E. BAKER ALL OF PEI ASSOC., INC.
C
C
C
C
      SUBROUTINE SH
      LOGICAL    KO, TVA
      INTEGER    BOTTOM, CLNLVL, CT, FMB, TYPCT
      REAL       IC, INDCST
      DIMENSION AC(13), ANN(25), BHP(2), CAPCST(10,8), CC(4), IAPCS(9),
     &          IC(6), INDCST(10), KO(2), P(500), TABLE(8,8)
      COMMON /COAL1/    HHV, PS, PA, CST, PNA, ALK, CL, FEO, PM
      COMMON /COSTS/  CAPCST, ANN, INDCST
      COMMON /FAN/    DELTAP, CSTFAN
      COMMON /INPUT/    BSIZE, CF, AHFX, AHXT, HHVROM, PSROM, PAROM, CSTROM,
     &                  PNAROM, ALKROM, CLROM, FEOROM, PMROM, HHVCLN, PSCLN,
     &                  PACLM, CSTCLN, PNACLN, ALKCLN, CLCLN, FEOCLN, PMCLN,
     &                  FMB, NR, IS, CT, TYPCT, CLNLVL, BOTTOM, NM, IAPCS, KO,
     &                  TVA
      COMMON /INSYSP/ CCEFF, UTEFF, BNHR, FANLOS, QIN, ALKIN
      COMMON /MATBAL/ TABLE
      COMMON /OPT/    BYPFGD, BYPFF, EFFESP, EFFLSD, EFFLIM
      COMMON /PARM/   P
      COMMON /ESC/ IYC, RATPLT, RATLA2, RATMA2, RATOM
      COMMON /STREAM/ PPH, ACFM, TEMP, PPHMST, PPHALK, PPHPRT, PPHSO2, PPHNOX
C
C
      IODB = 8
```

```
      IO6 = 6
      IO7 = 7
      PI = 3.14159
      DO 10 I=1,13
      AC(I) = 0.0
   10 CONTINUE
      DO 15 I=1,4
      CC(I) = 0.0
   15 CONTINUE
      DO 20 I=1,6
      IC(I) = 0.0
   20 CONTINUE
C
      WRITE (IODB,8000)
 8000 FORMAT (T2,'ENTERING SUBROUTINE SH.')
      WRITE (IODB,8010) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,
     &                  PPHNOX,CF,CAPCST(4,1),INDCST,ANN,
     &                  (P(I),I=141,170)
 8010 FORMAT (T10,'PPH,ACFM,TEMP,PPHMST,PPHALK=',5F14.4,
     &       /T10,'PPHPRT,PPHSO2,PPHNOX,CF,CAPCST(SH)=',5F14.4,
     &       /T10,'INDCST=',8F10.2,
     &       /T10,'ANN=',7F10.2,
     &       /T14,6F10.2,
     &       /T14,6F10.2,
     &       /T14,6F10.2,
     &       /T10,'P(241-170)=',10F10.2,
     &       /T21,10F10.2,
     &       /T21,10F10.2)
C
C*******************************************************************************
C
C     PART 1:  SIZING CALCULATIONS.
C
C     THIS PART CALCULATES SIZING PARAMETERS FOR USE LATER IN THE
C     PROGRAM.
C
C     CALCULATE WATER TO BE ADDED TO THE GAS STREAM.
C
      H2OADD = (TEMP-160.) * (PPH-PPHMST) / 4220.
C
C     WRITE DEBUG.
C
      WRITE (IODB,8100) H2OADD
 8100 FORMAT (T10,'H2OADD=',F12.2)
C
C*******************************************************************************
C
C     PART 2:  MODULE SPECIFIC OUTPUT.
C
C     THIS PART WRITES THE MODULE SPECIFIC OUTPUT FOR SPRAY HUMIDIFI-
C     CATION.
C
```

```
C     WRITE HEADER.
C
      IF (KO(1)) WRITE (IO6,6200)
      IF (KO(2)) WRITE (IO7,6200)
 6200 FORMAT (///T5,'SPRAY HUMIDIFICATION')
C
C     WRITE DETAIL MODULE SPECIFIC OUTPUT.
C
      IF (KO(1)) WRITE (IO6,6210) P(244),P(264)
      IF (KO(2)) WRITE (IO7,6210) P(244),P(264)
 6210 FORMAT (T10,'WATER USEAGE DEPENDS ON INLET TEMPERATURE.  30',
     &             ' DEG F APPROACH',
     &       /T10,'TO SATURATION IS USED.  SATURATION IS ASSUMED',
     &             ' AT 130 DEG F.',
     &       /T10,'ONLY 1 UNIT IS USED (NO REDUNDANCY).  SPARE PUMPS',
     &             ' ARE UTILIZED.',
     &       /T10,'A FACTOR OF ',F3.1,' TIMES THE THEORETICAL WATER ',
     &             'USEAGE IS USED TO',
     &       /T10,'INSURE COOLING WITH THE EXCESS RECYCLED.  THE ',
     &             'CHAMBER IS EQUIPPED',
     &       /T10,'WITH MIST ELIMINATORS.  AVERAGE PRESSURE DROP IS ',
     &            F3.0,' IN H20.')
C
C*******************************************************************************
C
C     PART 3:  DIRECT CAPITAL COST CALCULATIONS.
C
C     THIS PART CALCULATES THE DIRECT CAPITAL COST FOR THE SPRAY
C     HUMIDIFICATION SYSTEM.
C
C     CALCULATE THE COST OF THE CHAMBER (CC(1)).  THIS COST INCLUDES
C     THE DUCTING, SPRAYERS, MIST ELIMINATORS, AND INSULATION.  INPUT
C     PARAMETERS TO THIS CALCULATION ARE THE GAS VELOCITY (PARAMETER
C     241), DUCTWORK FABRICATION AND INSTALLATION COST (PARAMETER 242)
C     AND EXTRA FABRICATION COST FACTOR (PARAMETER 243).
C
      CC(1) = SQRT(ACFM/(P(241)*PI))*3930.1*P(242)*P(243) + 9.1746*
     &        SQRT(ACFM)
C
C     CALCULATE THE COST OF THE SURGE TANK (CC(2)).  NOTE SIZE IS THE
C     OVERALL TANK CAPACITY, N IS THE NUMBER OF TANKS, AND Q IS THE
C     ACTUAL SIZE OF EACH TANK.  INPUT PARAMETERS ARE WATER USAGE
C     FACTOR (PARAMETER 244), SURGE TANK RETENTION TIME (PARAMETER
C     245), MAXIMUM TANK SIZE (PARAMETER 246), AND AN INSTALLATION
C     FACTOR (PARAMETER 251).
C
      SIZE = H2OADD  * P(244) * P(245) / 62.4
      N = IFIX(SIZE/P(246)) + 1
      Q = SIZE / N
      CC(2) = P(251) * 56600. * N * (Q/10949.)**.68
C
C     CALCULATE THE COST OF FEED PUMPS (CC(3)).  NOTE, BHP IS THE
```

```
C     PUMP'S BRAKE HORSEPOWER AND WILL BE USED LATER IN THE ANNUAL
C     COSTS.  INPUT PARAMETERS ARE EXTRA PUMPAGE FACTOR (PARAMETER
C     247), PUMP EFFICIENCY (PARAMETER 248), PUMP HEAD SIZE (PARA-
C     METER 249), AN INSTALLATION FACTOR (PARAMETER 251), AND FEED
C     PUMP REDUNDANCY (PARAMETER 252).
C
      BHP(1) = (H2OADD*P(244)*P(247)*P(249)) / (3960.*(P(248)/100.)*
     &          8.33*60.)
      CC(3) = 14400. * P(251) * P(252) * (BHP(1)/85.)**.47
C
C     CALCULATE THE COST OF THE FRESH WATER PUMPS (CC(4)).  THE CAL-
C     CULATIONS ARE THE SAME AS ABOVE EXCEPT A DIFFERENT HEAD SIZE
C     (PARAMETER 250) AND REDUNDANCY (PARAMETER 253) ARE USED.
C
      BHP(2) = (H2OADD*P(247)*P(250)) / (3960.*(P(248)/100.)*
     &          8.33*60.)
      CC(4) = 14400. * P(251) * P(253) * (BHP(2)/85.)**.47
C
C     SUM DIRECT CAPITAL COSTS.
C
      CAPCST(4,1) = (CC(1) + CC(2) + CC(3) + CC(4)) * RATPLT*P(296)
C
C     WRITE DEBUG.
C
      WRITE (IODB,8300) CC,CAPCST(4,1)
 8300 FORMAT (T10,'CC=',4F12.2,
     &        /T10,'CAPCST(SH)=',F12.2)
C
C*******************************************************************************
C
C     PART 4:  INDIRECT CAPITAL COST CALCULATIONS.
C
C     THE INDIRECT CAPITAL COSTS ARE GENERALLY A PERCENTAGE OF THE
C     DIRECT CAPITAL COSTS OR DIRECT PLUS INDIRECT CAPITAL COSTS.
C     THE PERCENTAGES ARE RETAINED IN THE PARAMETER FILE (PARAMETERS
C     154 THRU 159).  EACH INDIRECT COST FOR EACH MODULE IS SUMMED
C     INTO THE ARRAY 'INDCST'.
C
C      CALCULATE ENGINEERING DESIGN AND SUPERVISION(TVA) OR
C      GENERAL FACILITIES(EPRI)
C
      IC(1) = P(255)/100. * CAPCST(4,1)
      INDCST(1) = INDCST(1) + IC(1)
C
C      CALCULATE A-E CONTRACTOR(TVA) OR ENG. AND HOME OFF. FEES (EPRI)
C
      IC(2) = P(254)/100. * CAPCST(4,1)
      INDCST(2) = INDCST(2) + IC(2)
C
C      CALCULATE CONSTRUCTION EXPENSE (TVA) PROJECT CONTINGENCY (EPRI)
C
      IC(3) = P(256)/100. * CAPCST(4,1)
```

```
      INDCST(3) = INDCST(3) + IC(3)
C
C      CALCULATE CONTRACTOR FEES (TVA) OR PROCESS CONTINGENCY (EPRI)
C
      IC(4) = P(257)/100. * CAPCST(4,1)
      INDCST(4) = INDCST(4) + IC(4)
C
C      SUM DIRECT AND INDIRECT CAPITAL COSTS TO HAVE A BASE FOR THE
C      REMAINDER OF THE INDIRECT COSTS.
C
      COST = CAPCST(4,1) + IC(1) + IC(2) + IC(3) + IC(4)
C
C      CALCULATE CONTINGENCY (TVA) OR SALES TAX (EPRI)
C
      IF (TVA) THEN
            IC(5) = P(258)/100. * COST
         ELSE
            IC(5) = P(258)/100. * CAPCST(4,1)
         ENDIF
      INDCST(5) = INDCST(5) + IC(5)
C
C     INCLUDE IC(5) IN RUNNING COST BASE TOTAL
C
      COST = COST + IC(5)
C
C      CALCULATE ROYALTIES (TVA) OR ROYALTY ALLOWANCE (EPRI)
C
      IC(6) = P(259)/100. * COST
      INDCST(6) = INDCST(6) + IC(6)
C
C      CALCULATE INTEREST DURING CONSTRUCTION (TVA)
C
      IF (TVA) THEN
        IC(8) = P(268)/100. * COST
        INDCST(8) = INDCST(8) + IC(8)
      ENDIF
C
C       CALCULATE ALLOWANCE FOR STARTUP AND MODIFICATIONS (TVA) OR
C                  INITIAL CATALYST (EPRI)
C
      IC(9) = P(269)/100. * COST
      INDCST(9) = INDCST(9) + IC(9)
C
C       CALCULATE LAND (TVA AND EPRI)
C
      IC(10) = P(181)
      INDCST(10) = INDCST(10) + IC(10)
C
C     WRITE DEBUG.
C
      WRITE (IODB,8400) IC,INDCST
 8400 FORMAT (T10,'IC=',6F12.2,
```

```
     &        /T10,'INDCST=',4F12.2,
     &        /T17,4F12.2)
C
C*****************************************************************************
C
C     PART 5:  ANNUAL COST CALCULATIONS.
C
C     ANNUAL COSTS ARE REPORTED FOR THE ENTIRE SYSTEM (ALL MODULES).
C     THIS PART CALCULATES THE INCREMENTAL COST FOR THIS MODULE.
C     MOST PARAMETERS ARE KEPT BY QUANTITY.  ALL VALUES ARE ON A PER
C     YEAR BASIS.
C
C     THE OPERATIONAL LABOR IS A FIXED VALUE (PARAMETER
C     260) REGARDLESS OF THE SIZE.
      AC(1) = P(260)
      ANN(1) = ANN(1) + AC(1)
C
C     CALCULATE ELECTRICITY REQUIREMENT IN KILOWATTHOURS.  THE ELEC-
C     TRICITY USE IS BASED ON THE PUMP HORSEPOWER.
C
      AC(4) = (BHP(1)+BHP(2)) * 1.2 * .7457 * 8760. * CF
      ANN(4) = ANN(4) + AC(4)
C
C     CALCULATE WATER REQUIREMENT IN 1000 GALLONS.
C
      AC(5) = H2OADD/(8.33*1000.) * 1.1 * 1.1 * 8760. * CF
      ANN(5) = ANN(5) + AC(5)
C
C     NO REHEAT REQUIRED.
C
C     NO REAGENTS OR CHEMICALS REQUIRED.
C
C     CALCULATE MAINTENANCE MATERIAL COST AS A PERCENTAGE (PARAMETER
C     263) OF THE DIRECT CAPITAL COSTS.
C
      AC(10) = P(262)/100. * CAPCST(4,1)
      ANN(10) = ANN(10) + AC(10)
C
C     NO FUEL COST PREMIUM.
C
C     NO WASTES PRODUCED.
C
C     WRITE DEBUG.
C
      WRITE (IODB,8500) AC,ANN
 8500 FORMAT (T10,'AC=',7F10.2,
     &        /T13,6F10.2,
     &        /T10,'ANN=',7F10.2,
     &       /T14,6F10.2
     &       /T14,6F10.2
     &       /T14,6F10.2)
```

```
C
C*************************************************************************
C
C      PART 6:  MATERIAL BALANCE CALCULATIONS.
C
C      AT THE END OF EACH MODULE, THE GAS STREAM PARAMETERS CONTAINED
C      IN THE COMMON 'STREAM' ARE ADJUSTED FOR THE AFFECTS OF THIS
C      MODULE.
C
C      CALCULATE NEW FLUE GAS ACFM.
C
       ACFM = ACFM*620./(460.+TEMP) + (H2OADD/(8.33*60.))*(359/492.)*
      &       620.
C
C      CALCULATE NEW FLUE GAS FLOW IN LBS/HR.
C
       PPH = PPH + H2OADD
C
C      THE NEW FLUE GAS TEMPERATURE IS FIXED AT 160 DEG F.
C
       TEMP = 160.
C
C      CALCULATE NEW FLUE GAS MOISTURE.
C
       PPHMST = PPHMST + H2OADD
C
C      NO CHANGE TO FLUE GAS ALKALINITY, PARTICULATES, SO2, OR NOX.
C
C      ADJUST OVERALL PRESSURE DROP FOR THIS MODULE.  PARAMETER 264 IS
C      THE INCREMENTAL PRESSURE DROP.
C
       DELTAP = DELTAP + P(264)
C
C      WRITE DEBUG.
C
       WRITE (IODB,8600) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,
      &                  PPHNOX
 8600  FORMAT (T10,'PPH,ACFM,TEMP,PPHMST=',4F14.4,
      &       /T10,'PPHALK,PPHPRT,PPHSO2,PPHNOX=',4F14.4)
C
C*************************************************************************
C
C      PART 7:  CONCLUSION.
C
       WRITE (IODB,8700)
 8700  FORMAT (T2,'LEAVING SUBROUTINE SH.')
C
       RETURN
       END
```

```
$DEBUG
C
C
C
C           INTEGRATED AIR POLLUTION CONTROL SYSTEM PROGRAM
C           ========== === ========= ======= ====== =======
C
C                          SCA CALCULATION ROUTINE
C                          --- ---------- -------
C
C
C       WRITTEN BY:    JAY PALMISANO
C                      PEI ASSOCIATES, INC.
C                      FEBRUARY, 1986
C
C
C           THIS ROUTINE CALCULATES SPECIFIC COLLECTION AREA (SCA)
C       FOR AN ELECTROSTATIC PRECIPITATOR (ESP). THE ALGORITHMS WERE
C       TAKEN ADAPTED FROM INFORMATION PROVIDED BY DR. LES SPARKS, USEPA
C       AEERL WITH PRINCIPAL ORIGIN IN RESEARCH BY THE SOUTHERN RESEARCH
C       INSTITUTE.
C
C
C
      SUBROUTINE SCACAL(SCA,RESIS)
      INTEGER FMB,CT,BOTTOM,TYPCT,CLNLVL
      LOGICAL KO(2),TVA
      REAL INDCST,SCALST(14),RESLST(7),MFNA,MFFE,MFT,MFCM,SPA(3,3)
      REAL*8 EFFMAT(7,14),EH,EL
      COMMON /COSTS/ CAPCST(10,8),ANN(25),INDCST(10)
      COMMON /COAL1/ HHV,PS,PA,CST,PNA,ALK,CL,FEO,PM
      COMMON /STREAM/ PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
     &ALKROM,CLROM,FEOROM,PMROM,HHVCLN,PSCLN,PACLN,CSTCLN,PNACLN,ALKCLN,
     &CLCLN,FEOCLN,PMCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,IAPCS(9),KO
     &,TVA
      COMMON /INSYSP/ CCEFF,UTEFF,BNHR,FANLOS,QIN,ALKIN
      COMMON /PARM/ P(500)
      COMMON /MATBAL/ TABLE(8,8)
      COMMON /OPT/ BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
      COMMON /FAN/ DELTAP,CSTFAN
      DATA SCALST
     &/100.,150.,200.,250.,300.,350.,400.,450.,500.,600.
     & ,700.,800.,1000.,2000./
      DATA RESLST/5.E9,1.E10,5.E10,2.E11,5.E11,2.E12,5.E12/
      DATA EFFMAT/
     &97.     ,96.4    ,95.1    ,91.8    ,88.2    ,81.8    ,75.8    ,
     &99.      ,98.6    ,98.0    ,96.2    ,93.9    ,89.8    ,85.5    ,
     &99.63    ,99.4    ,99.1    ,98.0    ,96.6    ,93.96   ,90.8    ,
     &99.84    ,99.7    ,99.5    ,98.9    ,98.0    ,96.3    ,93.99   ,
     &99.93    ,99.8    ,99.76   ,99.4    ,98.8    ,97.6    ,95.96   ,
     &99.97    ,99.9    ,99.87   ,99.7    ,99.3    ,98.4    ,97.2    ,
```

```
     &99.98  ,99.95  ,99.93  ,99.8   ,99.5   ,98.9   ,98.0   ,
     &99.992 ,99.99  ,99.96  ,99.9   ,99.7   ,99.3   ,98.7   ,
     &99.996 ,99.991 ,99.97  ,99.93  ,99.8   ,99.5   ,99.0   ,
     &99.999 ,99.996 ,99.99  ,99.97  ,99.91  ,99.77  ,99.5   ,
     &99.9995,99.998 ,99.996 ,99.99  ,99.96  ,99.89  ,99.7   ,
     &99.9999,99.999 ,99.998 ,99.995 ,99.98  ,99.94  ,99.8   ,
     &100.   ,100.   ,100.   ,100.   ,99.994 ,99.98  ,99.95  ,
     &100.   ,100.   ,100.   ,100.   ,100.   ,100.   ,99.999 /
      DATA SPA/4.74,4.85,4.85,28.39,28.39,28.39,8.67,9.71,10.59/
C
C
C
C*******************************************************************************
C
C      PART 1:  CALCULATE ASH RESISTIVITY(REF. SORI-EAS-85-841,11/85
C                                          R. BICKELHAUPT, PP6-1,13)
C
C     RESISTIVITY IN PRESENCE OF LIMB
C>>>> IF LIMB IS PRESENT IN THE SYSTEM, RESISTIVITY IS FIXED AT PARAMETER
C>>>> VALUE.
      DO 1 I=1,9
        IF(IAPCS(I) .EQ. 2) THEN
          RESIS= P(122)*1.E9
          GO TO 200
        ENDIF
    1 CONTINUE
C
C       VOLUME RESISTIVITY
C
C>>>  DETERMINE MOLE FRACTIONS FOR NA2O, FE2O3, AND TOTAL ASH COMPONENTS
C>>>  USE ALKALINITY TO DETERMINE CALCIUM + MAGNESIUM CONCENTRATION; USE
C>>>  MOLECULAR WEIGHT OF CALCIUM IN CALCULATION SINCE THIS IS THE
C>>>  PREDOMINANT ALKALINE COMPONENT.
      MFCM= (ALK-PNA)/56.
      IF(MFCM .LE. 0.) MFCM=0
      MFNA= PNA/29.88
      MFFE= FEO/159.7
      MFT= 1.23+MFFE+MFNA+MFCM
C>>>  ATOMIC CONCENTRATIONS USING MOLECULAR CONCENTRATIONS AND CATIONIC
C>>>  RATIOS. USE STANDARD VALUE FOR LITHIUM.
      ACLI= .066
      ACCM= MFCM/MFT*100.*.5
      IF(ACCM .LE. 0.) ACCM = 5.0
      ACNA= MFNA/MFT*100.*.666667
      ACFE= MFFE/MFT*100.*.4
C>>>> VOLUME RESISTIVITY, OHM-CM
      WRITE(8,*) ' MFNA,MFFE,MFCM',MFNA,MFFE,MFCM
      RESLOG= 8.9434-1.8916*ALOG10(ACLI+ACNA)-.9696*ALOG10(ACFE)+
     &        1.237*(ALOG10(ACCM) - .4)
C>>   CORRECT TO FIELD INTENSITY OF 12KV/CM
      RESLOG= RESLOG-.3
C>>   CORRECT FOR TEMPERATURE
```

```
      TK= .555556*(TEMP-32.)+273.15
C>>   RESISTIVIY @ 1000/TK = 2.4
      RV24= 10.**(RESLOG-6.935+4335.5/417.)
      RESISV= 10.**(RESLOG-6.935+4334.5/TK)
      WRITE(8,*) ' PNA,FEO,MFNA,MFFE,ACNA,ACFE,RESLOG,RESISV',
     &           PNA,FEO,MFNA,MFFE,ACNA,ACFE,RESLOG,RESISV
C
C       SURFACE RESISTIVITY
C
C>>   WATER CONCENTRATION, %
      CW= PPHMST/PPH*100.
C
      R1L= 10.7737 - 2.334*ALOG10(ACNA*ACLI)
      R2L= R1L + (CW - 9.0)*(-.128)
C      R3L= R2L + (.56 - .056)*2.58
      R4L= R2L - .3
C>> CALCULATE SURFACE RESISTIVITY LIMIT AT 385K
      RSOL=R4L + .4343*7.3895E-4*EXP(2.3033E3/385.)*CW
C>>>SURFACE RESISTIVITY
      RSL= RSOL - .4343*7.3895E-4*EXP(2.3033E3/TK)*CW
C>>   RESISTIVIY @ 1000/TK = 2.4
      RS24= 10.**(RSOL-.4343*7.3895E-4*EXP(2.3033E3/417.)*CW)
      RESISS= 10.**RSL
      WRITE(8,*) ' R1L,R2L,R4L,RSOL,RESISS',R1L,R2L,R4L,RSOL,RESISS
C
C        ACID RESISTIVITY (@ 4KV/CM)
C
      IF(ACCM .LT. 5.0 .AND. ACFE .LT. 1.0) THEN
        C1= 2.635
        C2= 5.0
      ELSE
        C1= .2915
        C2= 2.0502
      ENDIF
C>>>> COMBINED SURFACE/VOLUME RESISTIVITY (@ E= 12 KV/CM,CW=10%,TC=144)
      RSVL= ALOG10(RS24*RV24/(RS24+RV24)) + .24
      RAL1= C1 + .7669*RSVL
C     CALCULATE INTERCEPT FOR [SO3] = 4 PPMV
      B= RAL1 + C2*ALOG10(4.0)
C>>>  CALCULATE SO3 CONCENTRATION FROM SO2
      PPMSO2= PPHSO2/(ACFM*60.)*359.*(TEMP+460.)/520./64.*1.E6
      PPMSO3= PPMSO2*.001
C>>>> CORRECT RESISTIVITY TO ACTUAL SO3 PPM
      RAL2 = B - C2*ALOG10(PPMSO3)
      WRITE(8,*) ' RSVL,B,PPMSO2,RAL2=',RSVL,B,PPMSO2,RAL2
C>>>  DETERIMINE SLOPE BASED ON ASH COMPOSITION AND ACID CONCENTRATION
      IF(ACLI+ACNA .GT. 1.0 .AND. ACCM .GT. 5.0) THEN
        J1= 1
      ELSEIF(ACFE .LT. 1.0 .AND. ACCM .LT. 3.0) THEN
        J1= 2
      ELSE
        J1= 3
```

```
      ENDIF
      IF(PPMSO3 .LT. 2.75) THEN
        I1= 1
      ELSEIF(PPMSO3 .LT. 6.5) THEN
        I1= 2
      ELSE
        I1= 3
      ENDIF
C>>>> CORRECT RESISTIVITY TO ACTUAL TEMPERATURE AND E= 12 KV/CM
      B2= RAL2 + SPA(I1,J1)*2.4
      RAL3= (B2 - SPA(I1,J1)*1000./TK)*.76 + 1.95
      WRITE(8,*) ' RA3,B2,I1,J1,SPA(I1,J1)=',RAL3,B2,I1,J1,SPA(I1,J1)
      RESISA= 10.**RAL3
C
C     COMBINED SURFACE, VOLUME, AND ACID RESISTIVITY
C
      RESIS= RESISA*RESISS*RESISV/
     &       (RESISA*RESISV+RESISA*RESISS+RESISV*RESISS)
C
C
C*******************************************************************************
C
C     PART 2: CALCULATE SCA
C
  200 IF(RESIS .LT. RESLST(1)) THEN
        RESIS= RESLST(1)
      ELSEIF(RESIS .GT. RESLST(7)) THEN
        WRITE(6,*)' RESISTIVITY GREATER THAN MAXIMUM; FORCED TO MAXIMUM'
        RESIS= RESLST(7)
      ENDIF
C
C     DETERMINE SCA BY INTERPOLATION ON TABLE OF RESISTIVITY&EFFICIENCY
C
      DO 2 M=2,7
        IF(RESIS .LT. RESLST(M)) GO TO 3
    2 CONTINUE
      M= 7
    3 IF(EFFESP .LT. EFFMAT(M,1)) THEN
        SCA= SCALST(1)
        GO TO 99
      ELSE
        DO 4 N=2,14
          EH= EFFMAT(M,N)+(EFFMAT(M-1,N)-EFFMAT(M,N))*
     &    (1.-(RESIS-RESLST(M-1))/(RESLST(M)-RESLST(M-1)))
          EL= EFFMAT(M,N-1)+(EFFMAT(M-1,N-1)-EFFMAT(M,N-1))*
     &    (1.-(RESIS-RESLST(M-1))/(RESLST(M)-RESLST(M-1)))
          IF(EFFESP .LE. EH) GO TO 5
    4   CONTINUE
      ENDIF
      N= 14
    5 RR=(DBLE(EFFESP)-EL)/(EH-EL)
      SCA= SCALST(N-1)+(SCALST(N)-SCALST(N-1))*RR
```

```
      WRITE(8,*) ' M,N,EH,EL,SCA,RESIS,SCALST',  M,N,EH,EL,SCA,RESIS,
     &SCALST
C
C*************************************************************************
C
C       PART 4:  ROUND SCA TO NEAREST WHOLE NUMBER.
C
   99 SCA = IFIX(SCA+.5)
C
C*************************************************************************
C
C       PART 5: CONCLUSION.
C
      RETURN
      END
```

```
$DEBUG
$large
$nofloatcalls
C
C          INTEGRATED AIR POLLUTION CONTROL SYSTEM
C                      COST MODEL
C
C             WRITTEN BY: PEI ASSOC., INC. (JP)
C                         4-12-83
C
C
      CHARACTER*4 IY,OPTMZ,DBGT,NA1
      CHARACTER*13 NEXTFL,DBGF
      CHARACTER*80 COMM1,COMM2
      INTEGER FMB,CT,BOTTOM,TYPCT,CLNLVL,AHE,IYC
      LOGICAL KO(2),OPTIM,TVA,FGDF
      REAL INDCST,CIPU(4,11),NTRAIN
      DIMENSION HMOD(9),HSTRM(5,8),NST(9),HCAP(10,9),SHCAP(6,3,9),AC(13)
      DIMENSION STREQ(8),OPTEQ(5)
      COMMON/CSTNDX/ICEPYR,CEPNDX,INVYR,IREVYR,RATLAB,RATMAT,ARM,ARL
      COMMON /OPT/     BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
      COMMON /COSTS/ CAPCST(10,8),ANN(25),INDCST(10)
      COMMON /COAL1/ HHV,PS,PA,CST,PNA,ALK,CL,FEO,PM
      COMMON /STREAM/ PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
      COMMON /INPUT/BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,HHVCLN,PSCLN,PACLN,CSTCLN,PNACLN,ALKCLN,
     &CLCLN,FEOCLN,PMCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,IAPCS(9),KO
     &,TVA
      COMMON /INSYSP/ CCEFF,UTEFF,BNHR,FANLOS,QIN,ALKIN
      COMMON /PARM/ P(500)
      COMMON /COMM/ COMM1,COMM2
      EQUIVALENCE (PPH,STREQ(1)),(BYPFGD,OPTEQ(1))
      COMMON /MATBAL/ TABLE(8,8)
      COMMON /FAN/ DELTAP,CSTFAN
      COMMON/BIG1/BAGDLP,BAGRAT,BAGCOST,BGLIFE,PSSO2X,EBYPAS,RS,PSC,
     1 CCSA,EFFPS,ESPDLP,RESIST,SCARAT,FANHP,FANLC,FANMC,DELTA1,FILTSA,
     2 PSF,FILRAT,H,TL,VD,VE,TGASID,SKLAND,DRAINM,DRAINL,CLAYSK,WPSO3,
     3 WPSO4,WPCO3,WPINS,ASHCLN,WINDEX,PDMAX,FLOOD,SRAIR,OXAIR,OXSCFM,
     4 PRES,SO2RT,SO2RV,SO2RST,SDFEE,VPD1,VT,HVCLN,HPTONW,
     5 ISLUDG,IFIXS,JSR,IVPD,IT,LON,LGMAX,ISTAND,NSBR,
     6 KEPASS,KPASO2,ICLAR,JSSO,JDIS,JSTR,JGPM,IRH,NSBF
      COMMON/COAL/HVC,WPC,WPH,WPO,WPN,WPSUL,WPCL,WPASH,WPH2O,SULO,ASHO
      COMMON/FLYASH/IASH,ASHUPS,ASHSCR,ASHPRU,ASHPRS,ASHLBU,ASHLBS
      COMMON/HGAS2/MPCT,LBMH,LBH,SCFM,HHG,HWG,CFR,BCFR,RCFR
      COMMON/PCOST/POND(9),PDACRE,TOPHRS,PNDCAP,PNDLAB(9),PNDMAT(9),ELAN
      COMMON/PND1/PDEPTH,PMXEXC,DISTPD,ILINER,CLAYIN,CLAYUC,HYPLMC,
     1            HYPLLC,PSAMAX,ACREC,APRAIN,SAPILE
      COMMON/SCRUBR/LG,V1,PPMOUT,SO2R,ELIMIT,PR,PD1,WPS,OX,PH,NSTAGE,
     1              NGRID,HS,ACS,ACSTRN
      COMMON/SOLDIS/CASO3,CASO4,CACO3,INSOLS,WDIS,PSD,DENS,DEND
      COMMON/OPPSCH/IOPSCH,IYROP,IA(50)
      REAL MPCT(6),LBMH(6),LBH(6)
```

```
      COMMON/LANDIS/FILAND,TWLF,DISFUL,ELECLF,WATERF,FILANL,OVLF
      DIMENSION SW(4),AREA(10),B(5),ULFC(7)
      DIMENSION B1(5)
      REAL LG
      REAL INSOLS
      COMMON/COMPLE/ICOMPL,IRPT,IRUN,NEXT,IEQPR,IWTBAL,ITM,INPOPT,NOPART
      COMMON/PCOSTS/PCLEAR,PDEXEC,SODM,SODL,PERIML,PERIMM,ROADMM,ROADLL,
     1 RECLAM
      COMMON/PNDARW/AB,AID,AIE,AOD,AS,DL,P1,VEC,VETT,WELLM,WELLL,WPSITE
      COMMON/PNDIND/INDPND,PENGIN,PARCH,PFLDEX,PFEES,PCONT,PSTART,PCONIN
      COMMON/RRIND/XINT,PCTMNT,PDMNTP,XINFLA,PCTOVR,PCTADM,UNDCAP,
     1         PCTINS,UC,IECON
      REAL *8 XINT,PCTMNT,PDMNTP,XINFLA,PCTOVR,PCTADM,UNDCAP,PCTINS,UC(1
     1        8)
      DIMENSION CYD(7),TYD(7),FUELDT(7),BYD(5)
      COMMON/TAXES/ITAXFR,TXFRRT,TXRAT,FRRAT,SERVRT,ROYALT,IOTIME,OTRATE
      COMMON /TVAFAN/ TGAS,TWG,VPD,NTRAIN,DLTAPF,NOTRAN,NTOTAL,ISCRUB,
     & NOREDN,PD,DPRH
      COMMON /NEW/ IFACT,V,FC
      COMMON /ESC/ IYC,RATPLT,RATLA2,RATMA2,RATOM
      COMMON /YC/ IMC,IYA,IMA
      DATA IY/'Y'/
      DATA CIPU/261.2,312.0,222.7,90.3,297.0,323.9,242.4,95.0,
     &          314.0,336.2,263.9,100.0,316.9,337.6,265.1,105.0,
     &          324.4,345.9,265.5,110.3,340.6,363.2,278.8,115.8,
     &          357.7,381.4,292.7,121.6,375.5,400.4,307.3,127.6,
     &          394.3,420.4,322.7,134.0,414.0,441.5,338.9,140.7,
     &          434.7,463.5,355.8,147.7/
      DATA NA1/'N   '/
      OPEN (7,FILE='lpt1:')
      OPEN (6,FILE='con:')
      OPEN (5,FILE='con:')
      OPEN (13,FILE='PASSFILE.BIN',STATUS='OLD',FORM='BINARY')
      OPEN (14,FILE='PASS2.BIN',STATUS='NEW',FORM='BINARY')
      OPEN (17,FILE='OPTIMIZE.BIN',STATUS='OLD',FORM='BINARY')
      OPEN (55,FILE='FAILURE',STATUS='NEW')
C
CC    USER AND PARAMETER INPUT
C
      READ (13) (TABLE(1,J),J=1,8)
      READ (13) ANN(13)
      READ (13) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
      READ (13) CCEFF,UTEFF,BNHR,FANLOS,QIN,ALKIN
      READ (13) HHV,PS,PA,CST,PNA,ALK,CL,FEO,PM
      READ (13) BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,HHVCLN,PSCLN,PACLN,CSTCLN,PNACLN,ALKCLN,
     &CLCLN,FEOCLN,PMCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,
     &(IAPCS(I),I=1,9),KO,TVA
      READ (13) P
      READ (13) COMM1,COMM2
      READ(13) IFACT,V,FC
      READ (13) OPTMZ,NOPT,TARGET,NEXTFL
```

```
      READ(COMM1,'(A4)') DBGT
      IF(DBGT .NE. 'TEST') THEN
          DBGF= 'NUL           '
      ELSE
          DBGF= 'OUTPUT.DBG    '
      ENDIF
      open (8,file=DBGF,STATUS='NEW')
    5 CONTINUE
C
C     CALCULATE RATMAT,RATLAB,RATPLT,RATOM AND IYC (ESCALATION)
C
C>>>  DATE OF COSTS
      IF(P(199) .EQ. 0.) THEN
          IYC= 1982
          IMC= 6
      ELSE
          IF(P(199) .LT. 8206.) P(199)= 8206.
          IYC= INT(P(199))/100+1900
          IMC= MOD(INT(P(199)),100)
      ENDIF
      IF(P(200) .EQ. 0.) THEN
          IYA= 1982
          IMA= 6
      ELSE
          IF(P(200) .LT. 8206.) P(200)= 8206.
          IYA= INT(P(200))/100+1900
          IMA= MOD(INT(P(200)),100)
      ENDIF
C>>>  ESCALATION RATIOS
      RATMAT = RAT(2,IMC,IYC)
      RATLAB = RAT(3,IMC,IYC)
      RATPLT = RAT(1,IMC,IYC)
      RATOM = RAT(4,IMA,IYA)
      WRITE(8,*) ' RATS=',RATMAT,RATLAB,RATPLT,RATOM
      RATMA2 =RATMAT
      RATLA2 = RATLAB
C     5 CALL INPUT1
C
C     ZERO OPTIMIZATION VARIABLES
C
      DO 8 I=1,5
    8 OPTEQ(I)= 0.0
      READ(17) BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
C      IF(BYPFGD.NE.0..OR.BYPFF.NE.0..OR.EFFESP.NE.0..OR.EFFLSD.NE.0..OR.
C     & EFFLIM.NE.0.) OPTMZ=NA1
C
C     ZERO COST ARRAYS
C
   10 DO 20 I=1,10
      DO 20 J=1,8
   20 CAPCST(I,J)=0.0
      DO 30 I=1,25
```

```
   30 ANN(I)=0.0
      DO 40 I=1,10
   40 INDCST(I)=0.0
C       DO 45 I=1,8
C       DO 45 J=1,8
C    45 TABLE(I,J)= 0.0
      DELTAP=0.0
      CSTFAN=0.0
C
C     SET UNCONTROLLED STREAM CHARACTERISTICS
C
C>>>> SET COAL TO ROM CHARACTERISTICS
C       HHV=HHVROM
C       PS=PSROM
C       PA=PAROM
C       CST=CSTROM
C       PNA=PNAROM
C       ALK=ALKROM
C       CL=CLROM
C       FEO=FEOROM
C       PM=PMROM
C>>>> INITIAL STREAM AND SYSTEM PERFORMANCE CALCULATIONS
C       CALL UNCNTL
C      (STREQ IS COMMON STREAM)
C>>>> SET FIRST ENTRY IN MATERIAL BALANCE TABLE
C       DO 50 I=1,8
C    50 TABLE(1,I)=STREQ(I)
C
C     IF COAL IS CLEANED, RESET STREAM CHARACTERISTICS
C
C       IF(CLNLVL .EQ. 1) GO TO 60
C       HHV=HHVCLN
C       PS=PSCLN
C       PA=PACLN
C       CST=CSTCLN
C       PNA=PNACLN
C       ALK=ALKCLN
C       CL=CLCLN
C       FEO=FEOCLN
C       PM=PMCLN
C>>>> SAVE INITIAL NET HEAT RATE
C       BNHRSV=BNHR
C       CALL UNCNTL
C>>>> COMPUTE ENERGY PENALTY/BONUS DUE TO COAL CLEANING
C       CCEFF=BNHR-BNHRSV
C
C     PRE AIR HEATER EXIT CALCULATIONS
C
C>>>> SEARCH THROUGH PRE-AIR HEATER SYSTEM CALLING APPPROPRIATE SUBROUT.
   60 DO 70 AHE=1,3
C>>>  CHECK FOR POST AIR HEATER EXIT SUBROUTINE
      IF(IAPCS(AHE) .GT. 3) GO TO 80
```

```
C>>>  LOW NOX BURNERS/OVERFIRE AIR
      IF(IAPCS(AHE) .EQ. 1) CALL LNBOF
C>>>  LIMB
      IF(IAPCS(AHE) .EQ. 2) CALL LIMB
C**** NOTE: COAL CLEANING IS CONSIDERED ABOVE AND IGNORED HERE.
   70 CONTINUE
      AHE = 4
C>>>> SET SECOND ENTRY IN MATERIAL BALANCE TABLE
   80 DO 90 I=1,8
   90 TABLE(2,I)=STREQ(I)
C
C     POST AIR HEATER EXIT CALCULATIONS
C>>>> ARE ANY MODULES REMAINING?
      IF(AHE .GT. NM) GO TO 400
C>>>  FIND MODULE TYPE AND CALL APPROPRIATE SUBROUTINE
      DO 300 M=AHE,NM
      I=IAPCS(M)-3
      GO TO(100,110,120,130,140,150), I
C>>>  SPRAY HUMIDIFICATION
  100 CALL SH
      GO TO 200
C>>>  ESP
  110 CALL ESP
      GO TO 200
C>>>  FABRIC FILTER
  120 CALL FF
      GO TO 200
C>>>  LIME SPRAY DRYING
  130 CALL LSD
      GO TO 200
C>>>  LIMESTONE FGD
  140 CALL TVAFGD
      GO TO 200
C>>>  DRY SORBENT INJECTION
  150 CALL DSI
C
  200 CONTINUE
C>>>  SET APPROPRIATE ENTRY IN MATERIAL BALANCE TABLE
      IPOS= M-AHE+3
      DO 250J=1,8
  250 TABLE(IPOS,J)=STREQ(J)
C
  300 CONTINUE
C
C     IF THERE IS A PRESSURE DROP, COST FANS FOR SYSTEM.
C
C      IF(DELTAP .GT. 0.0) CALL FAN1
C
C     MAKE FINAL CALCULATIONS AND PRINT REPORTS
C
  400 CONTINUE
C  400 CALL OUTPUT
```

```
C
C     DOES USER WISH TO OPTIMIZE? IF SO, RESET APPROPRIATE EFFICIENCY
C     OR BYPASS PERCENT AND REDO.
C
C     'OPTIM' IS A LOGICAL FUNCTION SUBPROGRAM.
C      IF(OPTIM(0.)) GO TO 10
C
C
C
C
      FGDF= .FALSE.
      DO 403 I=1,NM
  403 IF(IAPCS(I) .EQ. 8) FGDF= .TRUE.
      IF (.NOT. FGDF) THEN
C
C     DISPOSAL COSTS
C
         WELLM=5000.
         WELLL=5100.
         RECLAM=3100.
         PCLEAR=1950.
         PDEXEC=1.75
         SODM=0.30
         SODL=0.49
         PERIML=6.05
         PERIMM=7.00
         ROADMM=6.00
         ROADLL=8.50
         CLAYUC = 0.0
         CLAYIN = 0.0
         HYPLMC = 0.0
         HYPLLC = 0.0
         CALL READIN
         CALL WPASS
         ENDIF
      WRITE (14) CAPCST, INDCST, ANN
      WRITE (14) HHV, PS, PA, CST, PNA, ALK, CL, FEO, PM
      WRITE (14) PPH, ACFM, TEMP, PPHMST, PPHALK, PPHPRT, PPHSO2, PPHNOX
      WRITE (14) BSIZE, CF, AHFX, AHXT, HHVROM, PSROM, PAROM, CSTROM, PNAROM,
     &ALKROM, CLROM, FEOROM, PMROM, HHVCLN, PSCLN, PACLN, CSTCLN, PNACLN, ALKCLN,
     &CLCLN, FEOCLN, PMCLN, FMB, NR, IS, CT, TYPCT, CLNLVL, BOTTOM, NM,
     &IAPCS, KO, TVA
      WRITE (14) CCEFF, UTEFF, BNHR, FANLOS, QIN, ALKIN
      WRITE (14) P
      WRITE (14) COMM1, COMM2
      WRITE (14) TABLE
      WRITE (14) DELTAP, CSTFAN
      WRITE (14) IYC, IMC, IYA, IMA, RATPLT, RATLA2, RATMA2, RATOM
      WRITE(14) IFACT, V, FC
      WRITE(14) BYPFGD, BYPFF, EFFESP, EFFLSD, EFFLIM
      WRITE (14) OPTMZ, NOPT, TARGET, NEXTFL
      CLOSE(14)
```

```
CLOSE(55, STATUS='DELETE')
END
```

```
C
C          INTEGRATED AIR POLLUTION CONTROL SYSTEM PROGRAM
C          ========== === ========== ======= ====== =======
C
C                     WASTE FIXATION ROUTINE
C                     ----- -------- -------
C
C
C
C     WRITTEN BY:  J. R. SCHOETTELKOTTE
C                  PEDCO COMPUTER SERVICES
C                  JUNE, 1983
C
C     WRITTEN FOR: G. E. BAKER
C                  PEI ASSOC., INC.
C                  PEI PN 3568-3
C                  J. O. MILLIKEN
C                  U.S. EPA; IERL-RTP
C
C
C          THIS ROUTINE CALCULATES DESIGN PARAMETERS AND DIRECT
C     CAPITAL COSTS FOR DSI'S WASTE FIXATION SYSTEM AND THE INDIRECT
C     AND ANNUAL OPERATING COSTS FOR THE ENTIRE DRY SORBENT INJECTION
C     SYSTEM.  THE DIRECT COSTS FOR THE DSI ITSELF AND THE MATERIAL
C     BALANCE WERE DONE EARLIER IN THE SUBROUTINE 'DSI'.  THE ALGO-
C     RITHMS WERE DEVELOPED BY B. A. LASEKE AND REVIEWED BY G. E.
C     BAKER BOTH OF PEI ASSOC., INC.
C          THE CALL STATEMENT VARIABLES 'SO2' AND 'ASH' ARE THE QUAN-
C     TITIES OF SO2 AND PARTICULATE (IN TONS PER YEAR) REMOVED BY THE
C     PARTICULATE CONTROL DEVICE, EITHER ESP OR FABRIC FILTER.  ALL
C     THE WASTES MUST BE FIXATED FOR NAHCOLITE CONTAMINATION.
C
C
      SUBROUTINE FIXATE (SO2, ASH)
      LOGICAL    KO, TVA
      INTEGER    BOTTOM, CLNLVL, CT, FMB, TYPCT
      REAL       IC, INDCST, NCTPH
      DIMENSION AC(13), ANN(25), CAPCST(10,8), IAPCS(9), IC(8), INDCST(10),
     &          KO(2), P(500), TABLE(8,8)
      COMMON /COAL1/    HHV, PS, PA, CST, PNA, ALK, CL, FEO, PM
      COMMON /COSTS/  CAPCST, ANN, INDCST
      COMMON /FAN/    DELTAP, CSTFAN
      COMMON /INPUT/   BSIZE, CF, AHFX, AHXT, HHVROM, PSROM, PAROM, CSTROM,
     &                 PNAROM, ALKROM, CLROM, FEOROM, PMROM, HHVCLN, PSCLN,
     &                 PACLM, CSTCLN, PNACLN, ALKCLN, CLCLN, FEOCLN, PMCLN,
     &                 FMB, NR, IS, CT, TYPCT, CLNLVL, BOTTOM, NM, IAPCS, KO,
     &                 TVA
      COMMON /ESC/ IYC, RATPLT, RATLA2, RATMA2, RATOM
      COMMON /INSYSP/ CCEFF, UTEFF, BNHR, FANLOS, QIN, ALKIN
      COMMON /MATBAL/ TABLE
```

```
      COMMON /OPT/     BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
      COMMON /PARM/    P
      COMMON /STREAM/ PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
C
C
      IODB = 8
      IO6 = 6
      IO7 = 7
      DO 10 I=1,13
      AC(I) = 0.
   10 CONTINUE
      DO 20 I=1,8
      IC(I) = 0.
   20 CONTINUE
C
      WRITE (IODB,8000)
 8000 FORMAT (T2,'ENTERING SUBROUTINE FIXATE.')
      WRITE (IODB,8010) CF,SO2,ASH,CAPCST(9,1),INDCST,ANN,(P(J),
     &                  J=271,290)
 8010 FORMAT (T10,'CF,SO2,ASH,CAPCST(DSI)=',4F14.2,
     &       /T10,'INDCST=',8F10.2,
     &       /T10,'ANN=',7F10.2,
     &       /T14,6F10.2,
     &       /T14,6F10.2,
     &       /T14,6F10.2,
     &       /T10,'P(271-290)=',10F10.2,
     &       /T21,10F10.2)
C
C*******************************************************************************
C
C     PART 1:  SIZING CALCULATIONS.
C
C     THIS PART CALCULATES SIZING PARAMETERS FOR USE LATER IN THE
C     ROUTINE.
C
C     CALCULATE THE HOURLY NAHCOLITE CONSUMPTION RATE OF THE DSI SY-
C     STEM.
C
      WRITE(IODB,*) SO2,P(275),P(271),P(272)
      NCTPH = SO2 * P(275)/100. * 1.3125 * P(271) * 1./(P(272)/100.)
C
C     DEBUG OUTPUT.
C
      WRITE (IODB,8100) NCTPH
 8100 FORMAT (T10,'NCTPH=',F14.4)
C
C*******************************************************************************
C
C     PART 2:  MODULE SPECIFIC OUTPUT.
C
C     MODULE SPECIFIC OUTPUT WAS HANDLED IN THE SUBROUTINE 'DSI'.
C
```

```
C*****************************************************************************
C
C     PART 3:  DIRECT CAPITAL COST CALCULATIONS.
C
C     THE DIRECT CAPITAL COST FOR THE DSI SYSTEM WAS CALCULATED IN
C     THE SUBROUTINE 'DSI'.  IN THIS PART, THE COST OF THE WASTE
C     TREATMENT EQUIPMENT IS CALCULATED.
C
C     THE COST OF THE WASTE TREATMENT EQUIPMENT IS BASED ON THE QUAN-
C     TITY OF SO2 AND ASH REMOVED BY THE PARTICULATE CONTROL DEVICE.
C
      CC = (7.914*SO2**.466 + 1.524*ASH**.4) * 10.**6
C
C     ADD TO THE DIRECT CAPITAL COST OF THE DSI SYSTEM TO GET THE
C     TOTAL DIRECT CAPITAL COST OF THE ENTIRE DSI MODULE.
C
      CAPCST(9,1) = CAPCST(9,1) + CC*RATPLT
C
C     DEBUG OUTPUT.
C
      WRITE (IODB,8300) CC,CAPCST(9,1)
 8300 FORMAT (T10,'CC,CAPCST(DSI)=',2F12.2)
C
C*****************************************************************************
C
C     PART 4:  INDIRECT CAPITAL COST CALCULATIONS.
C
C     THE INDIRECT CAPITAL COSTS ARE GENERALLY A PERCENTAGE OF THE
C     DIRECT CAPITAL COST OR DIRECT PLUS INDIRECT CAPITAL COSTS.  THE
C     PERCENTAGES ARE RETAINED IN THE PARAMETER FILE (PARAMETERS 276
C     THRU 281).  EACH INDIRECT COST FOR EACH MODULE IS SUMMED INTO
C     THE ARRAY 'INDCST'.
C
C      CALCULATE ENGINEERING DESIGN AND SUPERVISION(TVA) OR
C      GENERAL FACILITIES(EPRI)
C
      IC(1) = P(277)/100. * CAPCST(9,1)
      INDCST(1) = INDCST(1) + IC(1)
C
C      CALCULATE A-E CONTRACTOR(TVA) OR ENG. AND HOME OFF. FEES (EPRI)
C
      IC(2) = P(276)/100. * CAPCST(9,1)
      INDCST(2) = INDCST(2) + IC(2)
C
C      CALCULATE CONSTRUCTION EXPENSE (TVA) PROJECT CONTINGENCY (EPRI)
C
      IC(3) = P(278)/100. * CAPCST(9,1)
      INDCST(3) = INDCST(3) + IC(3)
C
C      CALCULATE CONTRACTOR FEES (TVA) OR PROCESS CONTINGENCY (EPRI)
C
      IC(4) = P(279)/100. * CAPCST(9,1)
```

```
      INDCST(4) = INDCST(4) + IC(4)
C
C      SUM DIRECT AND INDIRECT CAPITAL COSTS TO HAVE A BASE FOR THE
C      REMAINDER OF THE INDIRECT COSTS.
C
      COST = CAPCST(9,1) + IC(1) + IC(2) + IC(3) + IC(4)
C
C      CALCULATE CONTINGENCY (TVA) OR SALES TAX (EPRI)
C
      IF (TVA) THEN
           IC(5) = P(280)/100. * COST
         ELSE
           IC(5) = P(280)/100. * CAPCST(9,1)
         ENDIF
      INDCST(5) = INDCST(5) + IC(5)
C
C     INCLUDE IC(5) IN RUNNING COST BASE TOTAL
C
      COST = COST + IC(5)
C
C      CALCULATE ROYALTIES (TVA) OR ROYALTY ALLOWANCE (EPRI)
C
      IC(6) = P(281)/100. * COST
      INDCST(6) = INDCST(6) + IC(6)
C
C      CALCULATE INTEREST DURING CONSTRUCTION (TVA)
C
      IF (TVA) THEN
        IC(8) = P(288)/100. * COST
        INDCST(8) = INDCST(8) + IC(8)
      ENDIF
C
C       CALCULATE ALLOWANCE FOR STARTUP AND MODIFICATIONS (TVA) OR
C                 INITIAL CATALYST (EPRI)
C
      IC(9) = P(286)/100. * COST
      INDCST(9) = INDCST(9) + IC(9)
C
C       CALCULATE LAND (TVA AND EPRI)
C
      IC(10) = P(287)
      INDCST(10) = INDCST(10) + IC(10)
C
C     DEBUG OUTPUT.
C
      WRITE (IODB,8400) IC,INDCST
 8400 FORMAT (T10,'IC=',8F12.2,
     &       /T10,'INDCST=',8F12.2)
C
C*******************************************************************************
C
C     PART 5:  ANNUAL OPERATING COST CALCULATIONS.
```

```
C
C     ANNUAL OPERATING COSTS ARE REPORTED FOR THE ENTIRE SYSTEM (ALL
C     MODULES).  MOST PARAMETERS ARE KEPT BY QUANTITY.  ALL VALUES
C     ARE ON A PER YEAR BASIS.
C
C
      AC(1) = P(282)
      ANN(1) = ANN(1) + AC(1)
C
C     CALCULATE ELECTRICITY REQUIREMENT IN KILOWATTHOURS.
C
      AC(4) = (45.1*NCTPH + 55.18) * 8760. * CF
      ANN(4) = ANN(4) + AC(4)
C
C     NO WATER OR REHEAT REQUIRED FOR THE DSI.  SOME WATER IS NEEDED
C     IN THE FIXATION PROCESS BUT THAT IS INCLUDED IN THE WASTE DIS-
C     POSAL COST.
C
C     THE ONLY CHEMICAL TO BE COSTED IS NAHCOLITE.  LIME IS USED IN
C     THE FIXATION PROCESS BUT ITS COST IS ALSO INCLUDED IN THE WASTE
C     DISPOSAL COST.
C
      AC(9) = NCTPH * 8760. * CF
      ANN(9) = ANN(9) + AC(9)
C
      ANN(10) = ANN(10) + CAPCST(9,1) * P(284)*.01
C
C     ONLY WET WASTES ARE PRODUCED.  NOTE PARAMETER 273 IS THE PER-
C     CENT SOLIDS IN THE FIXATION WASTE STREAM.  A FIXATION COST
C     MULTIPLIER (PARAMETER 274) IS USED TO COVER THE ADDED COST OF
C     FIXATION.  HERE IT IS APPLIED TO THE WASTE QUANTITY PRODUCING
C     A QUANTITY PROPORTIONATELY HIGHER TO BE DISPOSED OF AT THE
C     REGULAR RATE.
C
      AC(12) = ASH/(P(273)/100.) * 8760 * CF * P(274)
      ANN(12) = ANN(12) + AC(12)
C
C     DEBUG OUTPUT.
C
      WRITE (IODB,8500) AC,ANN
 8500 FORMAT (T10,'AC=',7F12.2,
     &       /T13,6F12.2,
     &       /T10,'ANN=',7F12.2,
     &       /T14,6F12.2,
     &       /T14,6F12.2,
     &       /T14,6F12.2)
C
C*******************************************************************************
C
C     PART 6:  MATERIAL BALANCE CALCULATIONS.
C
C     MATERIAL BALANCE CALCULATIONS FOR THE DSI MODULE WERE DONE IN
```

```
C      THE SUBROUTINE 'DSI'.
C
C*******************************************************************************
C
C      PART 7:  CONCLUSION.
C
       WRITE (IODB,8700)
 8700 FORMAT (T2,'LEAVING SUBROUTINE FIXATE.')
C
       RETURN
       END
```

```
$DEBUG
C
      SUBROUTINE OPTIM(DUMMY)
C
C         THIS SUBPROGRAM DETERMINES IF COSTS ARE TO BE OPTIMEZED BY
C     LOWERING THE EFFICIENCY OF A CONTROL DEVICE. THIS IS
C     DONE EITHER DIRECTLY OR INDIRECTLY BY BYPASSING SOME OF THE GAS
C     STREAM. THE NEW EFFICIENCY OR BYPASS IS CALCULATED FROM A USER-
C     PROVIDED "TARGET" EMISSION RATE WHICH IS HIGHER THAN THAT RESULT-
C     ING FROM THE FIRST RUN.
C
C         WRITTEN BY: PEDCO ENVIRONMENTAL INC.
C                     6/83
C
C
      CHARACTER*4 Y,NA,AIAPCS,OPTMZ,BLANK,NA1,IO,HLP,YA1
      CHARACTER*13 NEXTFL,REC*79
      LOGICAL VALID,KO(2),TVA
      LOGICAL MOPT
      DIMENSION AIAPCS(8,9),MOPT(9),MN(5)
      COMMON /BATCHC/OPTMZ,NOPT,TARGET,NEXTFL
      COMMON /NEW/ IFACT,V,FC
      COMMON /COSTS/ CAPCST(10,8),ANN(25),INDCST(10)
      COMMON /COAL1/ HHV,PS,PA,CST,PNA,ALK,CL,FEO,PM
      COMMON /STREAM/ PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
      COMMON /INPUT/BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,HHVCLN,PSCLN,PACLN,CSTCLN,PNACLN,ALKCLN,
     &CLCLN,FEOCLN,PMCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,IAPCS(9),KO
      COMMON /INSYSP/ CCEFF,UTEFF,BNHR,FANLOS,QIN,ALKIN
      COMMON /PARM/ P(500)
      COMMON /MATBAL/ TABLE(8,8)
      COMMON /OPT/    BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
      DATA Y/'Y'/
      DATA NA/'N'/
      DATA NA1/'N   '/
      DATA YA1/'Y   '/
      DATA HLP/'H   '/
      DATA BLANK/'    '/
      DATA MOPT/.FALSE.,.TRUE.,2*.FALSE.,4*.TRUE.,.FALSE./
      DATA AIAPCS/
     &'LOW ','NOX ','BURN','ERS/','OVER','FIRE',' AIR','    ',
     &'LIMB',7*'    ',
     &'COAL',' CLE','ANIN','G   ',4*'    ',
     &'SPRA','Y HU','MIDI','FICA','TION',3*'    ',
     &'ESP ',7*'    ',
     &'FABR','IC F','ILTE','R   ',4*'    ',
     &'LIME','SPRA','Y DR','YING',4*'    ',
     &'LIME','STON','E FG','D   ',4*'    ',
     &'DRY ','SORB','ENT ','INJE','CTIO','N   ',2*'    '/
C
C     IF NO MODULE IS OPTIMIZIBLE, THEN RETURN.
C
```

```
      OPEN (15,FILE='OPTIMIZE.BIN',STATUS='NEW',FORM='BINARY')
      OPEN (16,FILE='TEST1.FIL',STATUS='NEW',ACCESS='SEQUENTIAL')
      OPEN (5,FILE='CON:')
      WRITE(8,*) 'NM,MOPT,OPTMZ',NM,MOPT,OPTMZ
      DO 10 I=1,NM
      IF (OPTMZ.NE.BLANK) THEN
         IF (OPTMZ.EQ.NA1) GOTO 999
         GOTO 4000
      ENDIF
      IF(MOPT(IAPCS(I))) GO TO 20
   10 CONTINUE
      GO TO 999
C
C     IF USER WISHES TO OPTIMIZE, DETERMINE WHAT DEVICE.
C
   20 WRITE(6,1000)
 1000 FORMAT(' DO YOU WANT TO OPTIMIZE? (Y/N/Help)')
      READ (5,10000,ERR=20) IO
10000 FORMAT(A1)
      IF(IO .EQ. NA .OR. IO .EQ. 'n') GO TO 999
      IF(IO .EQ. Y .OR. IO .EQ. 'y') GO TO 30
      IF(IO .EQ. HLP .OR. IO .EQ. 'h') CALL HELP('OPTHELP.DAT ')
      GO TO 20
   30 WRITE(6,1100)
 1100 FORMAT(/' THE FOLLOWING MODULE(S) MAY BE OPTIMIZED:')
      N= 0
      DO 40 I=1,NM
      IF(.NOT. MOPT(IAPCS(I))) GO TO 40
      N= N+1
      MN(N)= IAPCS(I)
      WRITE(6,1200) N,(AIAPCS(J,MN(N)),J=1,8)
 1200 FORMAT(I2,2H. ,8A4)
   40 CONTINUE
      WRITE(6,1300)
 1300 FORMAT(' ENTER INDEX NUMBER OF MODULE TO BE OPTIMIZED>')
      READ(5,*) L
      IF(L .LE. 0 .OR. L .GT. N) GO TO 20
      GOTO 99
C
C
C     BRANCH TO CALCULATION APPROPRIATE TO MODULE CHOSEN
C
 4000 IF(OPTMZ .EQ. YA1) THEN
         OPEN(88,FILE='BZ.XXX',STATUS='NEW')
         CLOSE(88)
      ENDIF
      IF (.NOT.MOPT(NOPT)) THEN
         WRITE (6,5000)
 5000 FORMAT(' OPTIMIZATION FAILED. MODULE CANNOT BE OPTIMIZED')
 5010 FORMAT(' OPTIMIZATION FAILED. TARGET EMMISION RATE TO LOW')
         GOTO 999
      ENDIF
```

```
      L=1
      MN(L)=NOPT
   99 CONTINUE
  100 IF(MN(L) .NE. 2) GO TO 200
C>>>> LIMB OPTIMIZATION
      ERSO2= PPHSO2/QIN
C>>>  GET NEW SO2 EMISSION
      IF (NOPT.NE.0) GOTO 1410
 1399 WRITE(6,1400) ERSO2
 1400 FORMAT(/' THE SO2 EMISSION RATE IS CURRENTLY ',F5.2,' LB/MMBTU.',
     &       /' ENTER NEW TARGET EMISSION RATE >')
      READ(5,*,ERR=1399) X
      GOTO 1420
C>>>  CALCULATE NEW EFFICIENCY
 1410 X=TARGET
 1420 IF(EFFLIM .NE. 0.) P(112)=EFFLIM
      EFFLIM= 100*(TABLE(1,7)/QIN - X)/(TABLE(1,7)/QIN)
      GO TO 900
  200 IF(MN(L) .NE. 5) GO TO 300
C>>>> ESP OPTIMIZATION
      ERPRT= PPHPRT/QIN
C>>>  GET NEW PARTICULATE EMISSION RATE
      IF (NOPT.NE.0) GOTO 1510
      WRITE(6,1500) ERPRT
 1500 FORMAT(' THE PARTICULATE EMISSION RATE IS CURRENTLY',F5.2,
     &' LB/MMBTU.'/' ENTER NEW EMISSION RATE (MUST BE HIGHER)>')
      READ(5,*) X
      GOTO 1520
 1510 X=TARGET
 1520 IF(X .LE. ERPRT) GO TO 200
C>>>  CALCULATE NEW EFFICIENCY
      EFFESP= 100.-(X/ERPRT*(100.-P(91)))
      GO TO 900
  300 IF(MN(L) .NE. 6) GO TO 400
C>>>> FABRIC FILTER
C>>   FIND FABRIC FILTER ENTRY IN TABLE (LOCFF)
      DO 310 I=1,3
      IF(IAPCS(I) .GT. 3) GO TO 320
  310 CONTINUE
      I= 4
C>    I-1 IS THE NUMBER OF PRE-AIR HEATER EXIT MODULES
  320 DO 330 J=1,NM
      IF(IAPCS(J) .EQ. 6) GO TO 340
  330 CONTINUE
  340 LOCFF= J-(I-1)+1
      ERPRT= (1.-P(72)*.009)*TABLE(LOCFF,6)/QIN
C>>>  GET NEW EMISSION RATE
      IF(NOPT.NE.0) GOTO 1550
 1503 WRITE(6,1505) ERPRT
 1505 FORMAT(T10,'AT LEAST 10% OF THE GAS STREAM MUST BE BYPASSED. IN'
     &      /T10,'ORDER TO ACHIEVE THIS, THE NEW EMISSION RATE MUST BE'
     &      /T9,F5.2,' LB/MMBTU OR MORE.'
```

```
     &      /T10,' PLEASE ENTER YOUR NEW TARGET EMISSION RATE>')
       READ(5,*,ERR=1503) X
       GOTO 1560
 1550 X=TARGET
 1560 IF(X .LE. ERPRT) GO TO 300
C>>>  CALCULATE BYPASS
C>>   CALCULATE FRACTION OF GAS TO BE BYPASSED
       BYPFF= 1.0-(1.0-X*QIN/(TABLE(LOCFF,6)))/(P(72)*.01)
C>>>  AT LEAST 10% OF THE GAS STREAM MUST BE BYPASSED
       IF(BYPFF .GE. P(73)*.01) GO TO 900
       WRITE(6,1600) BYPFF
 1600 FORMAT(/T5,' ** CALCULATED BYPASS FRACTION (',F4.3,') IS LESS THAN
     & 10%;',/T5,'     THIS IS INSUFFICIENT FOR ANY VALID OPTIMIZATION',
     &           ' TO BE PERFORMED.')
       BYPFF= 0.0
       IF (NOPT.NE.0) THEN
          WRITE (6,5010)
          GOTO 999
       ENDIF
       GO TO 20
C>>>> LIME SPRAY DRYING OPTIMIZATION
  400 IF(MN(L) .NE. 7) GO TO 500
       ERSO2= PPHSO2/QIN
C>>>  GET NEW EMISSION RATE
       IF (NOPT.NE.0) GOTO 1610
       WRITE(6,1400) ERSO2
       WRITE(6,1450)
 1450 FORMAT(1X,'(MUST BE HIGHER)')
       READ(5,*) X
       GOTO 1620
 1610 X=TARGET
 1620 IF(X .LE. ERSO2) GO TO 400
C>>>  CALCULATE NEW EFFICIENCY
       EFFLSD=100.-(X/ERSO2 * (100.-P(35)))
       GO TO 900
C>>>> LIMESTONE FGD OPTIMIZATION
C>>>  FIND FGD ENTRY IN TABLE
  500 DO 510 I=1,3
       IF(IAPCS(I) .GT. 3) GO TO 520
  510 CONTINUE
       I= 4
C>>   I-I IS THE NUMBER OF PRE-AIR HEATER EXIT MODULES
  520 DO 530 J=1,NM
       IF(IAPCS(J) .EQ. 8) GO TO 540
  530 CONTINUE
  540 LOCFGD=J-(I-1)+1
       ERSO2= PPHSO2/QIN
C>>>  GET TARGET EMISSION RATE
       IF (NOPT.NE.0) GOTO 1650
       WRITE(6,1505) ERSO2
       READ(5,*) X
       GOTO 1660
```

```
 1650 X=TARGET
 1660 IF(X .LE. ERSO2) X= ERSO2
C>>>  CALCULATE FRACTION OF THE GAS TO BE BYPASSED.
      BYPFGD=(X-(1.-P(203)*.01)*TABLE(1,7)/QIN)/
     &              (P(203)*.01*TABLE(1,7)/QIN)
C>>>  AT LEAST 1% OF THE GAS STREAM MUST BE BYPASSED.
      IF(BYPFGD .GE. 0.01) GO TO 900
      WRITE(6,1600) BYPFGD
      BYPFGD= 0.0
      IF (NOPT.NE.0) THEN
         WRITE (6,5010)
         GOTO 999
      ENDIF
      GO TO 20
C
C     RETURN TO RERUN SYSTEM WITH MODIFIED EFFICIENCY/BYPASS
C
  900 CONTINUE
      WRITE (15) BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
      CLOSE (15)
      RETURN
C
C     NO OPTIMIZATION;RETURN TO END PROGRAM.
C
  999 CLOSE (15,STATUS='DELETE')
      RETURN
C
      END
```

```
$LARGE
$DEBUG
C
C       THIS SUBROUTINE ACCEPTS USER INPUT INTO COMMON FOR USE BY THE
C     REMAINING ROUTINES. THE USER MAY EDIT THESE UPON COMPLETION OF
C     THE INPUT SEQUENCE. A DISK FILE CONTAINING SEMI-FIRM DESIGN & COST
C     PARAMETERS(ALSO USED THROUGHOUT THE PROGRAM) IS READ INTO COMMON
C     'PARM'. DEFINITIONS OF THESE PARAMETERS ARE CONTAINED WITHIN THE
C     FILE. AN INPUT SUMMARY IS ALSO PRINTED AS THE FIRST PART OF THE
C     SYSTEM'S OUTPUT.
C                WRITTEN BY: PEI ASSOC.,INC. (JP)
C                            (4-5-83)
C                     REVISED (12-17-84)  (JW)
C
      DIMENSION HMOD(9),HSTRM(5,8),NST(9),HCAP(10,9),SHCAP(6,3,9),AC(13)
      DIMENSION STREQ(8),OPTEQ(5),ICH(500)
      CHARACTER*4 AFMB,ACLNLV,ANR,ABOT,AIAPCS,ANU,ANL,AYU,AYL,OPTMZ,
     &  OPTINT,DBGT
      CHARACTER*12 FNAME,DBGF*13,INB*40
      CHARACTER*1 JSW,AA,JSW2
      CHARACTER*64 LAB(500)
      CHARACTER*80 COMM1,COMM2
      INTEGER*2 CW
      INTEGER FMB,CT,BOTTOM,TYPCT,CLNLVL
      LOGICAL VALID,KO(2),TVA,EX,EXB
      DIMENSION RINPEQ(26),INPEQ(17),TYPCC(6,22)
      DIMENSION AFMB(3,2),ACLNLV(8,2),ANR(2,2),ABOT(2),AIAPCS(8,9)
      COMMON /BATCHC/OPTMZ,NOPT,TARGET,NEXTFL
      CHARACTER*13 NEXTFL,NEXTIN,B12*12
      COMMON /OPT/     BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
      COMMON /MATBAL/ TABLE(8,8)
      COMMON /COMM/ COMM1,COMM2
      COMMON /COSTS/ CAPCST(10,8),ANN(25),INDCST(10)
      COMMON /STREAM/ PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
      COMMON/INSYSP/ CCEFF,UTEFF,BNHR,FANLOS,QIN,ALKIN
      COMMON /COAL1/HHV,PS,PA,CST,PNA,ALK,CL,FEO,PM
      COMMON/INPUT/BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,VROM,FCROM,HHVCLN,PSCLN,PACLN,CSTCLN,
     &PNACLN,ALKCLN,CLCLN,FEOCLN,PMCLN,VCLN,FCCLN,FMB,NR,IS,CT,TYPCT,
     &CLNLVL,BOTTOM,NM,IAPCS(9),KO,TVA
      COMMON /NEW/IFACT
      COMMON /PARM/ P(500)
      EQUIVALENCE (BSIZE,RINPEQ(1)),(FMB,INPEQ(1))
      EQUIVALENCE (PPH,STREQ(1)),(BYPFGD,OPTEQ(1))
C
C     ALPHANUMERIC TITLES FOR INPUT SUMMARY
C
      DATA AFMB/'    ','    ','WALL','TANG','ENTI','ALLY'/
      DATA ACLNLV/
     &'RUN-','OF-M','INE ','SORT','ED A','ND S','CREE','NED ',
     &'PHYS','ICAL',' CLE','ANIN','G   ',3*'    '/
      DATA ANR/'NEW ','    ','RETR','OFIT'/
```

```
      DATA ABOT/'DRY ','WET '/
      DATA AIAPCS/
     &'LOW ','NOX ','BURN','ERS/','OVER','FIRE',' AIR','    ',
     &'LIMB',7*'    ',
     &'COAL',' CLE','ANIN','G   ',4*'    ',
     &'SPRA','Y HU','MIDI','FICA','TION',' (SH',')   ','    ',
     &'ESP ',7*'    ',
     &'FABR','IC F','ILTE','R (F','F)  ',3*'    ',
     &'LIME','SPRA','Y DR','YING',' (LS','D)  ',2*'    ',
     &'LIME','STON','E FG','D (L','FGD)',3*'    ',
     &'DRY ','SORB','ENT ','INJE','CTIO','N (D','SI) ','    '/
      DATA JSW/'I'/
      DATA JSW2/'i'/
      DATA B12/'            '/
C
C     CHARACTERISTICS OF TYPICAL COALS (COAL TYPES 1-6 ACROSS,CHARACTER-
C     ISTICS (HHV,%SULFUR, ETC. ROM&CLEAN) DOWN)
C
      DATA TYPCC/ 11952.,11922.,12058.,10359., 8789., 7500.,
     &              2.23,  3.43,  0.89,  4.27,  0.56,  0.94,
     &              15.9,  13.0, 16.60, 20.60,  8.15,  5.90,
     &               0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
     &               0.4,   0.4,   0.4,   0.4,   2.8,   4.4,
     &               6.5,   6.5,   6.5,   6.5,  20.9,  31.9,
     &               0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
     &               9.0,   9.0,   9.0,   9.0,   9.0,   9.0,
     &               3.3,   5.0,   3.5,   9.6,  25.2,  32.0,
     &33.8,37.2,35.4,40.6,36.4,42.0,47.0,44.8,43.0,29.2,30.3,20.1,
     &             12596.,12845.,13611.,11507., 9050., 7840.,
     &              1.42,  2.74,  0.83,  3.50,  0.43,  0.54,
     &              10.0,   6.6,   4.6,  10.7,  6.46,   5.3,
     &             4.57,  3.85,  4.02,  3.74,  3.87,  3.82,
     &               0.4,   0.4,   0.4,   0.4,   2.8,   4.4,
     &               6.5,   6.5,   6.5,   6.5,  20.9,  31.9,
     &               0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
     &               9.0,   9.0,   9.0,   9.0,   9.0,   9.0,
     &               5.6,   4.4,   5.4,  11.0,  24.0, 30.01,
     &33.9,40.2,36.7,43.6,36.1,42.0,50.5,48.8,53.3,34.7,33.4,22.7/
C
      DATA ANU/'N'/,ANL/'n'/,AYU/'Y'/,AYL/'y'/
      DATA OPTINT/'    '/
      DATA NEXTIN/'              '/
      OPTMZ=OPTINT
      NEXTFL=NEXTIN
      TYPCT= 0
      CT= 0
      NOPT=0
      BYPFGD=0
      BYPFF=0
      EFFESP=0
      EFFLSD=0
      EFFLIM=0
```

```
C
C      OPEN FILES
C
       OPEN(7,FILE='LPT1:')
       OPEN(6,FILE='CON:')
       OPEN(5,FILE='CON:')
       OPEN(8,FILE='OUTINP.DBG',STATUS='NEW')
C>>>> OPEN FILE FOR MODULES PROGRAM
       OPEN(13,FILE='PASSFILE.BIN',STATUS='NEW',FORM='BINARY')
       OPEN (16,FILE='TEST.FIL',STATUS='NEW',ACCESS='SEQUENTIAL')
       OPEN (17,FILE='OPTIMIZE.BIN',STATUS='NEW',FORM='BINARY')
       OPEN (55,FILE='FAILURE',STATUS='NEW')
C>>>> CHECK FOR CHAINED BATCH FILE
       INQUIRE(FILE='NEXTRUN.BIN',EXIST=EX)
C>>>  IF CHAINED FILE DOES NOT EXIST, WAIT FOR USER RESPONSE
       IF (EX.EQV..FALSE.) GO TO 8000
       OPEN (1,FILE='NEXTRUN.BIN',STATUS='OLD',FORM='BINARY')
       READ (1) FNAME
        IF (FNAME.EQ.JSW.OR.FNAME.EQ.JSW2) GO TO 8000
       GO TO 8010
 8000 WRITE (6,4)
    4 FORMAT (' INPUT BATCH FILE NAME (FILENAME.EXT) OR I FOR ',
     1'INTERACTIVE >>')
       READ (5,9) FNAME
    9 FORMAT (1A12)
 8010 IF (FNAME.NE.JSW .AND. FNAME.NE.JSW2) THEN
          IF (FNAME .EQ. B12) GOTO 8000
          CALL BATCH(FNAME,EXB)
          IAPCS(NM+1)= 10
          IF(.NOT. EXB) THEN
             GO TO 8000
          ELSE
             GO TO 801
          ENDIF
       ENDIF
       CALL PARMED (ICH,LAB)
C
C      USER INPUT SEQUENCE
C
       WRITE(6,5)
    5 FORMAT(T16,'INTEGRATED AIR POLLUTION CONTROL COSTING PROGRAM',
      &       /T16,48('-')///)
   10 CONTINUE
   14 WRITE(6,17)
   17 FORMAT(' ENTER FIRST TITLE LINE FOR OUTPUT REPORTS',/' >')
       READ(5,'(A78)',ERR=14) COMM1
   15 WRITE(6,18)
   18 FORMAT(' ENTER SECOND TITLE LINE FOR OUTPUT REPORTS',/' >')
       READ(5,'(A78)',ERR=15) COMM2
   19 WRITE(6,20)
   20 FORMAT(' ENTER FIRING CONFIGURATION OF BOILER:',
     &  /T5,'1. WALL-FIRED ',
```

```
     &  /T5,'2. TANGENTIALLY FIRED >')
        READ(5,*,ERR=19)   FMB
        IF(FMB .NE. 1 .AND. FMB .NE. 2) GO TO 19
   30 WRITE(6,40)
   40 FORMAT(' ENTER BOILER SIZE IN MWE(100-1300)>')
        READ(5,*,ERR=30)   BSIZE
        IF (BSIZE .LT. 100. .OR. BSIZE .GT. 1300.) GO TO 30
   50 WRITE(6,60)
   60 FORMAT(' ENTER BOILER CAPACITY FACTOR (%)>')
        READ(5,*,ERR=50) CF
        CF= CF*.01
   70 WRITE(6,80)
   80 FORMAT(' ENTER IAPCS CONSTRUCTION STATUS(1=NEW,2=RETROFIT)>')
        READ(5,*,ERR=70) NR
        IF(NR .NE. 1 .AND. NR .NE. 2) GO TO 70
        WRITE (6,1001)
 1001 FORMAT (' COMMERCIAL OPERATION DATE IS USED IN LNB/OFA ONLY')
   90 WRITE(6,100)
  100 FORMAT(' ENTER YEAR OF COMMERCIAL OPERATION OF BOILER(YYYY)>')
        READ(5,*,ERR=90) IS
        IF(IS .LT. 1900 .OR. IS .GT. 3000) GO TO 90
  110 WRITE(6,120)
  120 FORMAT(' ENTER TEMPERATURE AT AIR HEATER EXIT (DEG. F.)>')
        READ(5,*,ERR=110) AHXT
C 130 WRITE(6,140)
C 140 FORMAT(' ENTER ACFM AT THE AIR HEATER EXIT;ENTER 0 TO CALCULATE>')
C       READ(5,*,ERR=130) AHFX
        AHFX= P(58)
  170 WRITE(6,180)
  180 FORMAT(' ENTER SELECTION OF TYPICAL COAL(1) OR SPECIFIC CHARACTERI
     &STICS(2)>')
        READ(5,*,ERR=170) I
        IF(I .NE. 1 .AND. I .NE. 2) GO TO 170
        GO TO (600,190), I
C
C       USER INPUT COAL CHARACTERISTICS
C
C>>>> ROM COAL
  190 CLNLVL=1
        WRITE(6,200)
  200 FORMAT(' ENTER COAL TYPE:',
     &       /T5,'1. BITUMINOUS',
     &       /T5,'2. SUB-BITUMINOUS',
     &       /T5,'3. LIGNITE >')
        READ(5,*,ERR=190)  CT
        IF(CT .LT. 1 .OR. CT .GT. 3) GO TO 190
        IFACT=14
        IF (CT.EQ.3) IFACT=18
  210 WRITE(6,220)
  220 FORMAT(' ENTER AVERAGE ROM COAL HIGHER HEATING VALUE(BTU/LB)>')
        READ(5,*,ERR=210)  HHVROM
  230 WRITE(6,240)
```

```
  240 FORMAT(' ENTER ROM COAL AVERAGE SULFUR CONTENT(WT %)>')
      READ(5,*,ERR=230) PSROM
  250 WRITE(6,260)
  260 FORMAT(' ENTER ROM COAL AVERAGE ASH CONTENT(WT %)>')
      READ(5,*,ERR=250) PAROM
  270 WRITE(6,280)
  280 FORMAT(' ENTER ROM COAL COST($/TON)>')
      READ(5,*,ERR=270) CSTROM
  286 WRITE(6,287)
      WRITE(6,288) P(92)
  287 FORMAT(' NA2O CONTENT IS SIGNIFICANT FOR LIGNITE COAL ONLY')
  288 FORMAT(' ENTER ROM NA2O CONTENT OF ASH(WT %);ZERO FOR DEFAULT',
     & F6.2,' >')
      READ(5,*,ERR=286) PNAROM
      IF (PNAROM.EQ.0) PNAROM=P(92)
  290 WRITE(6,300)
  300 FORMAT(' ENTER ROM AVERAGE ALKALINITY OF ASH(WT %)>')
      READ(5,*,ERR=290)  ALKROM
  310 WRITE(6,320)
  320 FORMAT(' ENTER ROM COAL AVERAGE CHLORINE CONTENT(WT %)>')
      READ(5,*,ERR=310) CLROM
  330 WRITE (6,340)
  340 FORMAT(' ENTER ROM AVERAGE FE2O3 CONTENT OF ASH(WT %)>')
      READ(5,*,ERR=330) FEOROM
      WRITE(6,345)
  345 FORMAT(' THE FOLLOWING THREE ENTRIES PLUS THE ASH CONTENT MUST TO'
     &       'TAL 100%:'/)
  350 WRITE(6,360)
  360 FORMAT(' ENTER ROM COAL MOISTURE CONTENT(WT %)>')
      READ(5,*,ERR=350)  PMROM
  362 WRITE (6,3650)
 3650 FORMAT (' THE NEXT 2 ITEMS MUST BE ENTERED. THEY ARE USED ONLY ',
     & 'FOR FGD')
      WRITE (6,365)
  365 FORMAT (' ENTER ROM VOLATILE MATTER CONTENT(WT %)>')
      READ (5,*,ERR=362) VROM
  364 WRITE(6,367)
  367 FORMAT(' ENTER ROM FIXED CARBON CONTENT(WT %)>')
      READ (5,*,ERR=364) FCROM
  369 WRITE(6,366)
  366 FORMAT(' DO YOU WISH COAL CLEANING?')
      READ(5,900,ERR=369) A
C>>>> CLEAN COAL
      IF(A .EQ. ANU.OR.A.EQ.ANL) GO TO 700
      IF(A .NE. AYU.AND.A.NE.AYL) GO TO 369
      CLNLVL= 2
  370 WRITE(6,380)
  380 FORMAT(/' ENTER CLEAN COAL HIGHER HEATING VALUE(BTU/LB)>')
      READ(5,*,ERR=370) HHVCLN
  390 WRITE(6,400)
  400 FORMAT(' ENTER CLEAN COAL AVERAGE SULFUR CONTENT(WT %)>')
      READ(5,*,ERR=390) PSCLN
```

```
  410 WRITE(6,420)
  420 FORMAT(' ENTER AVERAGE CLEAN COAL ASH CONTENT (WT %)>')
      READ(5,*,ERR=410) PACLN
  430 WRITE(6,440)
  440 FORMAT(' ENTER COST OF CLEANED COAL ($/TON)>')
      READ(5,*,ERR=430) CSTCLN
  446 WRITE (6,287)
      WRITE(6,448) P(92)
  448 FORMAT(' ENTER CLEAN NA2O CONTENT OF ASH(WT %);ZERO FOR DEFAULT',
     & F6.2' >')
      READ(5,*,ERR=446) PNACLN
      IF (PNACLN.EQ.0) PNACLN=P(92)
  450 WRITE(6,460)
  460 FORMAT(' ENTER CLEAN AVERAGE ALKALINITY OF ASH(WT %)>')
      READ(5,*,ERR=450) ALKCLN
  470 WRITE(6,480)
  480 FORMAT(' ENTER CLEAN COAL AVERAGE CHLORINE CONTENT (WT %)>')
      READ(5,*,ERR=470) CLCLN
  490 WRITE(6,500)
  500 FORMAT(' ENTER CLEAN AVERAGE FE203 CONTENT OF ASH(WT %)>')
      READ(5,*,ERR=490) FEOCLN
      WRITE(6,345)
  510 WRITE(6,520)
  520 FORMAT(' ENTER CLEAN COAL MOISTURE CONTENT (WT %)>')
      READ(5,*,ERR=510) PMCLN
  522 WRITE (6,3650)
      WRITE (6,525)
  525 FORMAT (' ENTER CLEAN COAL VOLATILE MATTER CONTENT (WT %)>')
      READ (5,*,ERR=522) VCLN
  526 WRITE(6,527)
  527 FORMAT(' ENTER CLEAN COAL FIXED CARBON CONTENT (WT %)>')
      READ(5,*,ERR=526) FCCLN
      GO TO 700
C
C     TYPICAL COAL TYPES
C
  600 WRITE(6,610)
  610 FORMAT(' ENTER COAL CHOICE:'/
     &       '   1. BITUMINOUS - PENNSYLVANIA'/
     &       '   2. BITUMINOUS - OHIO'/
     &       '   3. BITUMINOUS - WEST VIRGINIA'/
     &       '   4. BITUMINOUS - ILLINOIS'/
     &       '   5. SUB-BITUMINOUS - WYOMING'/
     &       '   6. LIGNITE - NORTH DAKOTA >')
      READ(5,*,ERR=600) TYPCT
      IF(TYPCT .LT. 1 .OR. TYPCT .GT. 6) GO TO 600
      IFACT=14
      IF (TYPCT.EQ.6) IFACT=18
C
C     SET TYPICAL COAL CHARACTERISTICS
C
      DO 625 I=5,26
```

```
  625 RINPEQ(I)= TYPCC(TYPCT,I-4)
  620 WRITE(6,630)
  630 FORMAT(' ENTER COAL CLEANING LEVEL:'/
     &        '    1. RUN-OF-MINE SORTED AND SCREENED'/
     &        '    2. PHYSICAL COAL CLEANING >')
      READ(5,*,ERR=620) CLNLVL
  700 WRITE(6,710)
  710 FORMAT(' ENTER BOILER BOTTOM ASH CONFIGURATION:'/
     &        '    1. DRY-BOTTOM'/
     &        '    2. WET-BOTTOM>')
      READ(5,*,ERR=700) BOTTOM
  711 WRITE(6,706)
  706 FORMAT(' SELECT OUTPUT OPTION:'
     &        /'      1. OUTPUT TO PRINTER',
     &        /'      2. OUTPUT TO DISPLAY SCREEN',
     &        /'      3. BOTH ABOVE >')
      READ(5,*,ERR=711) IO
      IF(IO .NE. 2) KO(2)=.TRUE.
      IF(IO .GE. 2) KO(1)=.TRUE.
  720 CONTINUE
      CALL CLS
      WRITE(6,730)
C  730 FORMAT(' SELECT IAPCS CONFIGURATION OPTIONS FROM THE FOLLOWING:'/
C     &  '    NOX                  PARTICULATE                  SO2'/
C     &  '   -----                -------------                -----'/
C     &  ' 1. LOW NOX BURNERS/ 3. COAL CLEANING        2. LIMB'/
C     &  '    OVERFIRE AIR     4. SPRAY HUMIDIF. (SH)  3. COAL CLEANING'/
C     &  ' 2. LIMB             5. ESP                  4. SPRAY HUMIDIF'/
C     &  '                     6. FABRIC FILTER (FF)   7. LIME SPRAY'/
C     &  '                                                DRY (LSD)'/
C     &  '                                             8. LIMESTONE/LIME'
C     &/ '                                                FGD  (FGD)'/
C     &  '                                             9. DRY SORBENT'/
C     & '                                             INJECTION (DSI)'/)
  730 FORMAT(' SELECT IAPCS CONFIGURATION OPTIONS FROM THE FOLLOWING:'/
     &  '    MODULE                                     POLLUTANT(S)'/
     &  '   --------                                   --------------'/
     &  '  1. LOW NOX BURNERS/OVERFIRE AIR              NOX'/
     &  '  2. LIMB                                      NOX, SO2'/
     &  '  3. COAL CLEANING                             PART,SO2'/
     &  '  4. SPRAY HUMIDIFICATION (SH)                 PART,SO2'/
     &  '  5. ESP                                       PART'/
     &  '  6. FABRIC FILTER (FF)                        PART'/
     &  '  7. LIME SPRAY DRYING (LSD)                   SO2'/
     &  '  8. LIMESTONE/LIME FGD (FGD)                  SO2'/
     &  '  9. DRY SORBENT INJECTION (DSI)               SO2')
      WRITE(6,725)
  725 FORMAT(/,' THE FOLLOWING RULES APPLY TO SELECTING A ',
     & 'CONFIGURATION:'
     &/'  1 - METHOD 4 MAY NOT BE USED WITH METHODS 7 OR 9',/
     &'  2 - METHODS 5 OR 6 MAY NOT PRECEED (BUT MAY FOLLOW) 7 OR 9',/
     &'  3 - METHODS MUST BE IN ASCENDING NUMERICAL ORDER(EXCEPT AS IN',
```

```
     &/'       2 ABOVE)',/
     &'  4 - METHODS MAY NOT BE REPEATED IN THE SAME SYSTEM.'/
     & ,'  (GENERALLY THE POST COMBUSTION MODULES FOLLOW THE GAS PATH)')
  745 WRITE(6,750)
  750 FORMAT(' ENTER OPTION NUMBERS IN ORDER(SEPARATE BY COMMAS)'/
     &       ' >')
      INB= '                                                  '
      DO 3 I=1,9
    3 IAPCS(I)= 0
      READ(5,'(A40)',ERR=745) INB
      READ(INB,*,ERR=745,END=752) (IAPCS(I),I=1,9)
  752 DO 753 I=1,9
        IF(IAPCS(I) .EQ. 0) GO TO 754
  753 CONTINUE
  754 NM= I-1
C
C     CHECK IAPCS VALIDIDTY USING FUNCTION 'VALID'
C
      IAPCS(NM+1)=10
C
C     MODULE 3 (COAL CLEANING) MUST BE SELECTED IF COAL CLEANING
C     WAS PREVIOUSLY INDICATED
C
      IF(CLNLVL .EQ. 2) THEN
        DO 756 I=1,NM
          IF(IAPCS(I) .EQ. 3) GOTO 757
  756   CONTINUE
        PAUSE' YOU MUST SELECT COAL CLEANING AS A MODULE. '
        GO TO 720
      ELSE
        IF(CLNLVL .NE. 1) CLNLVL= 1
      ENDIF
  757 IF(.NOT. VALID(DUM)) THEN
        WRITE(6,758)
  758   FORMAT(' YOU HAVE SELECTED AN INVALID CONFIGURATION WHICH WILL',
     &         ' LEAD TO UNPREDICTABLE',/' RESULTS.  DO YOU WISH TO ',
     &         'CONTINUE(Y/N)>')
        READ(5,900) AA
        IF(AA .NE. AYU.AND.AA.NE.AYL) GO TO 720
      ENDIF
      CALL CLS
  755 WRITE(6,760)
  760 FORMAT(' EDIT ABOVE DATA (Y/N)?')
      READ(5,900,ERR=755) A
      IF(A .EQ. AYU.OR.A.EQ.AYL) GO TO 763
      IF(A .EQ. ANU.OR.A.EQ.ANL) GO TO 799
      GO TO 755
C
C     EDIT OF USER INPUT
C
  763 WRITE(6,766) (I,RINPEQ(I),I=1,2),(I,RINPEQ(I),I=4,10)
  766 FORMAT(//,I3,'.  BOILER SIZE= ',F10.1,
```

```
     &         /,I3,'. CAPACITY FACTOR= ',F4.2,
     &         /,I3,'. AIR HEATER EXIT TEMPERATURE= ',F10.2,
     &         /,I3,'. ROM HHV= ',F10.1,
     &         /,I3,'. ROM SULFUR CONTENT= ',F5.2,
     &         /,I3,'. ROM ASH CONTENT= ',F5.2,
     &         /,I3,'. ROM COAL COST= ',F7.2,
     &         /,I3,'. ROM NA2O CONTENT OF ASH= ',F5.2,
     &         /,I3,'. ROM ALKALINITY OF ASH= ',F5.2)
  765 WRITE(6,764)
  764 FORMAT(' << HIT [RETURN] TO CONTINUE LISTING >>')
      READ(5,'(A1)',ERR=765) AA
      WRITE(6,768) (I,RINPEQ(I),I=11,26)
  768 FORMAT( I3,'. ROM CHLORINE CONTENT= ',F5.2,
     &         /,I3,'. ROM FE2O3 CONTENT OF ASH= ',F5.2,
     &         /,I3,'. ROM MOISTURE CONTENT= ',F5.2,
     &         /,I3,'. ROM VOLATILE MATTER= ',F5.2,
     &         /,I3,'. ROM FIXED CARBON=',F5.2,
     &         /,I3,'. CLEAN HHV= ',F10.1,
     &         /,I3,'. CLEAN SULFUR CONTENT= ',F5.2,
     &         /,I3,'. CLEAN ASH CONTENT= ',F5.2,
     &         /,I3,'. CLEAN COAL COST= ',F7.2,
     &         /,I3,'. CLEAN NA2O CONTENT OF ASH= ',F5.2,
     &         /,I3,'. CLEAN ALKALINITY OF ASH= ',F5.2,
     &         /,I3,'. CLEAN CHLORINE CONTENT= ',F5.2,
     &         /,I3,'. CLEAN FE2O3 CONTENT OF ASH= ',F5.2,
     &         /,I3,'. CLEAN MOISTURE CONTENT= ',F5.2,
     &         /,I3,'. CLEAN VOLATILE MATTER=',F5.2,
     &         /,I3,'. CLEAN FIXED CARBON=',F5.2
     &         )
  767 WRITE(6,764)
      READ(5,'(A1)',ERR=767) AA
      WRITE(6,770) (I,INPEQ(I-26),I=27,34)
  770 FORMAT( I3,'. FIRING CONFIGURATION= ',I1,
     &         /,I3,'. CONSTRUCTION STATUS= ',I1,
     &         /,I3,'. START DATE= ',I4,
     &         /,I3,'. COAL TYPE= ',I1,
     &         /,I3,'. TYPICAL COAL TYPE= ',I1,
     &         /,I3,'. CLEANING LEVEL= ',I1,
     &         /,I3,'. BOTTOM ASH CONFIGURATION= ',I1,
     &         /,I3,'. NUMBER OF CONTROL OPTIONS= ',I1)
      NJ=34+NM
      WRITE(6,722) (I,INPEQ(I-26),(AIAPCS(J,INPEQ(I-26)),J=1,8),I=35,NJ)
  722 FORMAT(//,' SYSTEM CONFIGURATION (CONTROL MODULE NUMBERS)',
     &     9(  /,I3,'. = ',I1,2X,8A4))
  777 WRITE(6,780)
  780 FORMAT(' CHANGE ANY OF THE ABOVE(Y/N)?')
      READ(5,900,ERR=777) A
      IF(A .EQ. ANU.OR.A.EQ.ANL) GO TO 799
  782 WRITE(6,785)
  785 FORMAT(' ENTER INDEX NUMBER TO CHANGE>')
      READ(5,*,ERR=782) I
      IF(I .GE. 40) GO TO 782
```

```
  786 WRITE(6,787)
  787 FORMAT(' ENTER NEW VALUE>')
      READ(5,*,ERR=786) V
      IF(I .GT. 26) INPEQ(I-26)=V
      IF(I .LE. 26) RINPEQ(I)=V
      GO TO 763
C
  799 CONTINUE
C
C     PRINT INPUT SUMMARY
C
C>>>> LOOP ON OUTPUT DEVICE NUMBER
  801 DO 800 LD=6,7
      IF(.NOT. KO(LD-5)) GO TO 800
C>>>  GENERAL HEADINGS
      WRITE(LD,10000) COMM1,COMM2
C>>>> BOILER INFORMATION
      WRITE(LD,11000) BSIZE,(AFMB(I,FMB),I=1,3),
     &                ABOT(BOTTOM),CF,AHXT,IS,(ANR(I,NR),I=1,2)
C      IF(AHFX.EQ.0.)WRITE(LD,11001) BSIZE,(AFMB(I,FMB),I=1,3),
C     &                ABOT(BOTTOM),CF,AHXT,IS,(ANR(I,NR),I=1,2)
C>>>  COAL INFORMATION
      IA=5+11*(CLNLVL-1)
      IB=IA+10
      WRITE(LD,12000) (ACLNLV(I,CLNLVL),I=1,8),(RINPEQ(I),
     &                I=IA,(IA+3)),RINPEQ(IA+6),(RINPEQ(I),I=
     &                (IA+8),IB)
      WRITE(LD,12050) RINPEQ(IA+4),RINPEQ(IA+5),RINPEQ(IA+7)
C>>>  IAPCS CONFIGURATION
      WRITE(LD,13000)(J,(AIAPCS(I,IAPCS(J)),I=1,8),J=1,NM)
C
C>>>  ECONOMIC PREMISES
      WRITE(LD,14000)
      IF(TVA) THEN
         WRITE(LD,14010)
      ELSE
         WRITE(LD,14020)
      ENDIF
C
      IF (LD .EQ. 6) WRITE(6,1400)
      IF (LD .EQ. 7) WRITE(6,1500)
C
  800 CONTINUE
      GOTO 9999
C
C  INPUT FORMATS
C
  900 FORMAT(A1)
  920 FORMAT(I2)
C
C     INPUT SUMMARY FORMATS
C
```

```
 1400 FORMAT(///)
 1500 FORMAT(1H1)
10000 FORMAT(//T14,
     &'INTEGRATED AIR POLLUTION CONTROL SYSTEM COSTING PROGRAM',/T13,
     & 57(1H-),//1X,A78/1X,A78//T33,'USER INPUT SUMMARY',/T31,21(1H-))
11000 FORMAT(//T11,'BOILER SIZE:',F10.0,' MW',T37,3A4,' FIRED, ',A4
     & ,' BOTTOM',
     &       /T11,'CAPACITY FACTOR:',2PF4.1,' %',21X,0PF8.0,' DEG.F'
     &      //T11,'DATE OF COMMERCIAL OPERATION OF BOILER:',1X,I4,
     &       /T11,'CONSTRUCTION STATUS OF CONTROL SYSTEM:',1X,2A4)
11001 FORMAT(//T11,'BOILER SIZE:',F10.0,' MW',T37,3A4,' FIRED, ',A4
     & ,' BOTTOM',
     &       /T11,'CAPACITY FACTOR:',2PF4.1,' %',
     &       /T11,'FLOW RATE AT AIR HEATER EXIT:',' TO BE CALCULATED',
     &            '  ',F8.0,' DEG.F'
     &      //T11,'DATE OF COMMERCIAL OPERATION OF BOILER:',1X,I4,
     &       /T11,'CONSTRUCTION STATUS OF CONTROL SYSTEM:',1X,2A4)
12000 FORMAT(//T11,'COAL CLEANING LEVEL: ',8A4,
     &       /T11,'COAL CHARACTERISTICS AT THIS CLEANING LEVEL:',
     &       /T54,'HHV (BTU/#):',F7.1,
     &       /T47,'SULFUR CONTENT (%):',F5.2,
     &       /T50,'ASH CONTENT (%):',F5.2,
     &       /T53,'COST ($/TON):',F5.2,
     &      /T45,'CHLORINE CONTENT (%):',F5.2,
     &       /T45,'MOISTURE CONTENT (%):',F5.2,
     &       /T38,'VOLATILE MATTER CONTENT (%):',F5.2,
     &       /T41,'FIXED CARBON CONTENT (%):',F5.2)
12050 FORMAT(//T11,' ASH CHARACTERISTICS AT THIS CLEANING LEVEL:',
     &       /T49,'NA20 CONTENT (%):',F5.2,
     &       /T51,'ALKALINITY (%):',F5.2,
     &       /T48,'FE203 CONTENT (%):',F5.2)
13000 FORMAT(//T11,'CONTROL SYSTEM CONFIGURATION:',
     &      9(/T43,I1,' - ',8A4))
14000 FORMAT(//T11'ECONOMIC PREMISES (TVA/EPRI): ')
14010 FORMAT(1H+,T41,'TVA')
14020 FORMAT(1H+,T41,'EPRI')
C
 9999 CONTINUE
C>>   PRINT PARAMETER FILE CHANGES
C
C     OPEN DEBUG FILE (IF ANY)
C
      READ(COMM1,'(A4)') DBGT
      IF(DBGT .NE. 'TEST') THEN
          DBGF= 'NUL           '
      ELSE
          DBGF= 'OUTINP.DBG    '
      ENDIF
      open (8,file=DBGF,STATUS='NEW')
      CALL PDUMP(LAB)
C
C     SET UNCONTROLLED STREAM CHARACTERISTICS
```

```
C
C>>>> SET COAL TO ROM CHARACTERISTICS
      HHV=HHVROM
      PS=PSROM
      PA=PAROM
      CST=CSTROM
      PNA=PNAROM
      ALK=ALKROM
      CL=CLROM
      FEO=FEOROM
      PM=PMROM
      VC=VROM
      FC=FCROM
C>>>> INITIAL STREAM AND SYSTEM PERFORMANCE CALCULATIONS
      CALL UNCNTL
C     (STREQ IS COMMON STREAM)
C>>>> SET FIRST ENTRY IN MATERIAL BALANCE TABLE
      DO 51 I=1,8
   51 TABLE(1,I)=STREQ(I)
C
C     IF COAL IS CLEANED, RESET STREAM CHARACTERISTICS
C
      IF(CLNLVL .EQ. 1) GO TO 61
      HHV=HHVCLN
      PS=PSCLN
      PA=PACLN
      CST=CSTCLN
      PNA=PNACLN
      ALK=ALKCLN
      CL=CLCLN
      FEO=FEOCLN
      PM=PMCLN
      VC=VCLN
      FC=FCCLN
C>>>> SAVE INITIAL NET HEAT RATE
      BNHRSV=BNHR
      CALL UNCNTL
      CCEFF=BNHR-BNHRSV
61    CONTINUE
C
C     STORE VARIABLES FOR MODULES PROGRAM
C
      WRITE (13) (TABLE(1,J),J=1,8)
      WRITE (13) ANN(13)
      WRITE (13) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
      WRITE (13) CCEFF,UTEFF,BNHR,FANLOS,QIN,ALKIN
      WRITE (13) HHV,PS,PA,CST,PNA,ALK,CL,FEO,PM
      WRITE (13) BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,HHVCLN,PSCLN,PACLN,CSTCLN,PNACLN,ALKCLN,
     &CLCLN,FEOCLN,PMCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,
     &(IAPCS(I),I=1,9),KO,TVA
      WRITE (13) P
```

```
      WRITE (13) COMM1,COMM2
      WRITE (13) IFACT,VC,FC
      WRITE (13) OPTMZ,NOPT,TARGET,NEXTFL
      CLOSE(13)
      WRITE (17) BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
      CLOSE(17)
      WRITE (16,*) (TABLE(1,J),J=1,8)
      WRITE (16,*) ANN(13)
      WRITE (16,*) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
      WRITE (16,*) CCEFF,UTEFF,BNHR,FANLOS,QIN,ALKIN
      WRITE (16,*) HHV,PS,PA,CST,PNA,ALK,CL,FEO,PM
      WRITE (16,*) BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,HHVCLN,PSCLN,PACLN,CSTCLN,PNACLN,ALKCLN,
     &CLCLN,FEOCLN,PMCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,
     &(IAPCS(I),I=1,9),KO,TVA
      WRITE (16,*) P
      WRITE (16,*) IFACT,VC,FC
      WRITE (16,*) BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
      CLOSE(16)
C>>>  FLAG SUCCESSFUL COMPLETION
      CLOSE(55,STATUS='DELETE')
      END
```

```
C
C       INTEGRATED AIR POLLUTION CONTROL SYSTEM PROGRAM
C       ========== === ========= ======= ====== =======
C
C                DRY SORBENT INJECTION (DSI) ROUTINE
C                --- ------- --------- ----- -------
C
C
C
C   WRITTEN BY:    J. R. SCHOETTELKOTTE
C                  PEDCO COMPUTER SERVICES
C                  JULY, 1983
C
C   WRITTEN FOR:   G. E. BAKER
C                  PEI ASSOC., INC.
C                  PEI PN 3568-3
C                  J. O. MILLIKEN
C                  U.S. EPA; IERL-RTP
C
C
C
C       THIS ROUTINE CALCULATES DESIGN PARAMETERS, SOME CAPITAL
C   COSTS AND MATERIAL BALANCE FOR DRY SORBENT INJECTION (DSI).
C   DSI REQUIRES WASTE FIXATION WHICH REQUIRES KNOWLEDGE OF THE
C   PARTICULATE DEVICE'S OPERATION.  MOST OF THE COSTING IS DONE
C   IN THE SUBROUTINE 'FIXATE' WHICH IS CALLED FROM SUBROUTINES
C   'ESP' OR 'FF' ONLY IF PRECEEDED BY DSI.  THE ALGORITHMS WERE
C   DEVELOPED BY B. A. LASEKE AND REVIEWED BY G. E. BAKER BOTH OF
C   PEI ASSOC., INC.
C
C
C
      SUBROUTINE DSI
      LOGICAL KO, TVA
      INTEGER    BOTTOM, CLNLVL, CT, FMB, TYPCT
      REAL       INDCST
      DIMENSION ANN(25), CAPCST(10,8), IAPCS(9), INDCST(10), KO(2), P(500),
     &           TABLE(8,8)
      COMMON /COAL1/   HHV, PS, PA, CST, PNA, ALK, CL, FEO, PM
      COMMON /COSTS/  CAPCST, ANN, INDCST
      COMMON /FAN/    DELTAP, CSTFAN
      COMMON /INPUT/   BSIZE, CF, AHFX, AHXT, HHVROM, PSROM, PAROM, CSTROM,
     &                PNAROM, ALKROM, CLROM, FEOROM, PMROM, HHVCLN, PSCLN,
     &                PACLM, CSTCLN, PNACLN, ALKCLN, CLCLN, FEOCLN, PMCLN,
     &                FMB, NR, IS, CT, TYPCT, CLNLVL, BOTTOM, NM, IAPCS, KO,
     &                TVA
      COMMON /INSYSP/ CCEFF, UTEFF, BNHR, FANLOS, QIN, ALKIN
      COMMON /MATBAL/ TABLE
      COMMON /ESC/ IYC, RATPLT, RATLA2, RATMA2, RATOM
      COMMON /OPT/    BYPFGD, BYPFF, EFFESP, EFFLSD, EFFLIM
      COMMON /PARM/   P
      COMMON /STREAM/ PPH, ACFM, TEMP, PPHMST, PPHALK, PPHPRT, PPHSO2, PPHNOX
```

```
C
C
      IODB = 8
      IO6 = 6
      IO7 = 7
C
      WRITE (IODB,8000)
 8000 FORMAT (T2,'ENTERING SUBROUTINE DSI.')
      WRITE (IODB,8010) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,
     &                  PPHNOX,BSIZE,CF,PS,CAPCST(9,1),INDCST,ANN,
     &                  (P(J),J=271,290)
 8010 FORMAT (T10,'PPH,ACFM,TEMP,PPHMST,PPHALK=',5F14.4,
     &       /T10,'PPHPRT,PPHSO2,PPHNOX,BSIZE,CF,PS=',6F14.4,
     &       /T10,'CAPCST(DSI)=',F10.2,
     &       /T10,'INDCST=',8F10.2,
     &       /T10,'ANN=',7F10.2,
     &       /T14,6F10.2,
     &       /T14,6F10.2,
     &       /T14,6F10.2,
     &       /T14,6F10.2,
     &       /T10,'P(271-290)=',10F10.2,
     &       /T21,10F10.2)
C
C*******************************************************************************
C
C     PART 1:  SIZING CALCULATIONS.
C
C     NO SIZING CALCULATIONS FOR THIS MODULE.
C
C*******************************************************************************
C
C     PART 2:  MODULE SPECIFIC OUTPUT.
C
C     THIS PART WRITES THE MODULE SPECIFIC OUTPUT FOR THE DRY SOR-
C     BENT INJECTION.
C
C     WRITE HEADER.
C
      IF (KO(1)) WRITE (IO6,6200)
      IF (KO(2)) WRITE (IO7,6200)
 6200 FORMAT (//T5,'DRY SORBENT INJECTION.')
C
C     WRITE DETAIL MODULE-SPECIFIC OUTPUT.
C
      IF (KO(1)) WRITE (IO6,6210) P(285),P(275)
      IF (KO(2)) WRITE (IO7,6210) P(285),P(275)
 6210 FORMAT (T10,'DRY SORBENT INJECTION IS A CONTINUOUS, NAHCOLITE ',
     &            'BASED SYSTEM',
     &       /T10,'CURRENTLY SET AT A NORMALIZED STOICHOMETRIC RATIO ',
     &            '(NSR) OF ',F3.1,'.',
     &       /T10,'DSI WILL RESULT IN AN SO2 REMOVAL OF ',F4.1,'% ',
     &            'WHEN USED WITH A 1.5%',
```

```
     &       /T10,'SULFUR COAL AND A FABRIC FILTER FOR PARTICULATE ',
     &            'CONTROL.')
C
C*************************************************************************
C
C     PART 3:  DIRECT CAPITAL COST CALCULATIONS.
C
C     THIS PART CALCULATES THE DIRECT CAPITAL COSTS FOR THE DRY SOR-
C     BENT INJECTION SYSTEM.  THE CAPITAL COST OF THE WASTE TREATMENT
C     SYSTEM IS COMPUTED IN THE 'FIXATE' ROUTINE.
C
C     THE COST OF THE DSI IS BASED ON BOILER SIZE AND COAL SULFUR
C     CONTENT.
C
      CAPCST(9,1) = 15.27 * EXP(.41*PS) * (BSIZE/500.)**.6 * 1000. *
     &              BSIZE * RATPLT * P(297)
C
C     DEBUG OUTPUT.
C
      WRITE (IODB,8300) CAPCST(9,1)
 8300 FORMAT (T10,'CAPCST(DSI)=',F10.2)
C
C*************************************************************************
C
C     PART 4:  INDIRECT CAPITAL COST CALCULATIONS.
C
C     INDIRECT CAPITAL COSTS ARE CALCULATED FOR THE ENTIRE DSI MODULE
C     IN SUBROUTINE 'FIXATE'.
C
C*************************************************************************
C
C     PART 5:  ANNUAL OPERATING COST CALCULATIONS.
C
C     ANNUAL OPERATING COSTS ARE CALCULATED FOR THE ENTIRE DSI MODULE
C     IN SUBROUTINE 'FIXATE'.
C
C*************************************************************************
C
C     PART 6:  MATERIAL BALANCE CALCULATIONS.
C
C     AT THE END OF EACH MODULE, THE STREAM PARAMETERS CONTAINED IN
C     THE COMMON 'STREAM' ARE ADJUSTED FOR THE AFFECTS OF THIS
C     MODULE.
C
C     CALCULATE THE NEW FLUE GAS PARTICULATES.  RETAIN THE CURRENT
C     VALUE AS 'OPHPRT' FOR USE LATER.
C
      OPHPRT = PPHPRT
      PPHPRT = OPHPRT + 1.3125*P(271)*(1./(P(272)/100.))*PPHSO2
C
C     CALCULATE NEW FLUE GAS FLOW (LBS/HR).
C
```

```
      PPH = PPH + PPHPRT - OPHPRT
C
C     CALCULATE NEW FLUE GAS ALKALINITY.
C
      PPHALK = PPHALK + 1.3125*P(271)*(1./(P(272)/100.))*PPHSO2
C
C     NO CHANGE TO FLUE GAS ACFM, TEMPERATURE, MOISTURE, SO2, AND
C     NOX.
C
C     NO ADJUSTMENT TO THE SYSTEM PRESSURE DROP IS REQUIRED.
C
C     DEBUG OUTPUT.
C
      WRITE (IODB,8600) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,
     &                  PPHNOX
 8600 FORMAT (T10,'PPH,ACFM,TEMP,PPHMST=',4F14.4,
     &       /T10,'PPHALK,PPHPRT,PPHSO2,PPHNOX=',4F14.4)
C
C*******************************************************************************
C
C     PART 7:  CONCLUSION.
C
      WRITE (IODB,8700)
 8700 FORMAT (T2,'LEAVING SUBROUTINE DSI.')
C
      RETURN
      END
```

```
C
C                INTEGRATED AIR POLLUTION CONTROL SYSTEM PROGRAM
C                ========== === ======== ======= ====== =======
C
C                        LIME SPRAY DRYING COST ROUTINE
C                        ---- ----- ------ ---- -------
C
C
C     WRITTEN BY:  J. R. SCHOETTELKOTTE
C                  PEDCO COMPUTER SERVICES
C                  MAY, 1983
C
C     WRITTEN FOR:  G. E. BAKER
C                   PEI ASSOC., INC.
C                   PEI PN 3568-3
C                   J. O. MILLIKEN
C                   U.S. EPA; IERL-RTP
C
C
C          THIS ROUTINE CALCULATES DESIGN PARAMETERS, CAPITAL AND
C     ANNUAL OPERATING COSTS, AND MATERIAL BALANCE FOR LIME SPRAY
C     DRYERS (LSD).  THE ALGORITHMS WERE DEVELOPED BY M. D. GIORDANO
C     AND REVIEWED BY G. E. BAKER BOTH OF PEI ASSOC., INC.
C
C
C
      SUBROUTINE LSD
      LOGICAL    KO, TVA
      INTEGER    BOTTOM, CLNLVL, CT, FMB, TYPCT
      REAL       IC, INDCST, LG, LMPPH, MLMPPH, LMTEMP
      DIMENSION AC(13), ANN(25), CAPCST(10,8), CC(18), IAPCS(9), IC(8),
     &          INDCST(10), KO(2), P(500), TABLE(8,8)
      COMMON /ESC/ IYC, RATPLT, RATLA2, RATMA2, RATOM
      COMMON /COAL1/    HHV, PS, PA, CST, PNA, ALK, CL, FEO, PM
      COMMON /COSTS/  CAPCST, ANN, INDCST
      COMMON /FAN/    DELTAP, CSTFAN
      COMMON /INPUT/   BSIZE, CF, AHFX, AHXT, HHVROM, PSROM, PAROM, CSTROM,
     &                 PNAROM, ALKROM, CLROM, FEOROM, PMROM, HHVCLN, PSCLN,
     &                 PACLM, CSTCLN, PNACLN, ALKCLN, CLCLN, FEOCLN, PMCLN,
     &                 FMB, NR, IS, CT, TYPCT, CLNLVL, BOTTOM, NM, IAPCS, KO,
     &                 TVA
      COMMON /INSYSP/ CCEFF, UTEFF, BNHR, FANLOS, QIN, ALKIN
      COMMON /MATBAL/ TABLE
      COMMON /OPT/    BYPFGD, BYPFF, EFFESP, EFFLSD, EFFLIM
      COMMON /PARM/   P
      COMMON /STREAM/ PPH, ACFM, TEMP, PPHMST, PPHALK, PPHPRT, PPHSO2, PPHNOX
C
C
      IODB = 8
      IO6 = 6
```

```
      IO7 = 7
      NOWAY = 0
      DO 10 I=1,13
      AC(I) = 0.
   10 CONTINUE
      DO 15 I=1,18
      CC(I) = 0.
   15 CONTINUE
      DO 20 I=1,8
      IC(I) = 0.
   20 CONTINUE
C
      WRITE (IODB,8000)
 8000 FORMAT (T2,'ENTERING SUBROUTINE LSD')
      WRITE (IODB,8010) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,
     &                  PPHNOX,EFFLSD,BSIZE,CF,(CAPCST(7,I),I=1,4),
     &                  INDCST,ANN,(P(J),J=31,60)
 8010 FORMAT (T10,'PPH,ACFM,TEMP,PPHMST,PPHALK=',5F14.4,
     &       /T10,'PPHPRT,PPHSO2,PPHNOX,EFFLSD,BSIZE,CF=',6F14.4,
     &       /T10,'CAPCST(LSD)=',4F10.2,
     &       /T10,'INDCST=',8F10.2,
     &       /T10,'ANN=',7F10.2,
     &       /T14,6F10.2,
     &       /T14,6F10.2,
     &       /T14,6F10.2,
     &       /T10,'P(31-60)=',10F10.2,
     &       /T19,10F10.2,
     &       /T19,10F10.2)
C
C*******************************************************************************
C
C     PART 1:  SIZING CALCULATIONS.
C
C     THIS PART CALCULATES THE PRELIMINARY VALUES REQUIRED FOR COST-
C     ING THE LSD.
C
C     CHECK THE INCOMING FLUE GAS TEMPERATURE.  IF THE TEMPERATURE
C     IS LESS THAN THE LSD SPRAY DOWN TEMPERATURE (PARAMETER 40),
C     LSD CAN NOT BE USED.  SET VARIABLE 'NOWAY' AND SKIP CALCULA-
C     TIONS.
C
      IF (TEMP .GT. P(40)) GO TO 100
      NOWAY = 1
      GO TO 200
C
C     ASSIGN EFFICIENCIES.  'EFF1' STARTS AS THE MAXIMUM EFFICIENCY
C     FOR LSD (PARAMETER 35).  'EFFLSD' IS EQUAL TO 'EFF1' UNLESS
C     THIS IS AN OPTIMIZATION, IN WHICH CASE 'SO2EF' HAS ALREADY BEEN
C     ASSIGNED.
C
  100 EFF1 = P(35)
      IF (EFFLSD .GT. 0.9*EFF1) EFFLSD = EFF1
```

```
      IF (EFFLSD .EQ. 0.) EFFLSD = EFF1
C
C     CALCULATE BYPASS FACTOR.
C
  110 BYPFAC = EFFLSD/EFF1
C
C     DETERMINE REACTIVE ALKALINITY OF THE FLYASH.  THE PERCENT REAC-
C     TIVE ALKALINITY IS BASED ON COAL TYPE (VARIABLE CT).  THEY ARE
C     CONTAINED IN THE PARAMETER FILE (PARAMETERS 49, 50, AND 51).
C
      RECALK = ALKIN * P(48+CT)
C
C     CALCULATE THE LIME USAGE RATE.
C
      LMPPH = BYPFAC * (PPHSO2 * (P(31)/64.) - ((P(32)/100.) * (RECALK/
     &        P(33))*0.085)) * P(34) * 56. * (EFF1/P(35))
C
C     ADJUST LIME USAGE FOR POTENTIAL LIME IN THE FLUE GAS FROM LIMB
C     WHICH CAN BE USED BY THE LIME SPRAY DRYER.
C
      LMTEMP= LMPPH
      ALLIMB = PPHALK - ALKIN
      LMPPH = LMPPH - ALLIMB
      IF(LMPPH .LT. 0.) LMPPH= 0.
C
C     CALCULATE WATER USAGE RATE.  THE WATER IS USED TO INTRODUCE THE
C     LIME AS A SLURRY AND TO REDUCE THE GAS TEMPERATURE.
C
      WATPPH = (TEMP-P(40)) * ((PPH-PPHMST)/4220.)
C
C     CALCULATE THE PERCENT OF SOLIDS IN THE SLURRY.
C
      PCTSLD = (LMPPH * (1. + (P(35)/(EFF1*(P(34)/(1-P(34))))))) +
     &         PPHPRT) * (100./WATPPH)
C
C     CALCULATE THE ALKALINITY EFFICIENCY.
C
      ALKEFF = P(38) * (RECALK/(P(37)*BSIZE))
C
C     IF THE PERCENT SOLIDS IS GREATER THAN THE MAXIMUM PERCENT
C     SOLIDS (PARAMETER 36), THE SOLID PERCENTAGE MUST BE LOWERED
C     BY REDUCING EFFICIENCY.
C
      IF (PCTSLD .LE. P(36)) GO TO 120
      IF(LMPPH .GT. 0.) GO TO 115
      NOWAY= 3
      GO TO 200
C
C     CALCULATE MODIFIED LIME USAGE RATE 'MLMPPH' AND NEW EFFICIENCY
C     'EFF1'.  THEN RETURN TO RECALCULATE BYPASS FACTOR AND LIME USAGE
C     RATE.
C
```

```
  115 MLMPPH = LMPPH * (1. - ((PCTSLD-P(36))/PCTSLD))
      EFF1 = P(35) * (((MLMPPH/LMPPH) * (1./(P(35)/(P(35)-ALKEFF)))) +
     &       ALLIMB/LMTEMP*75.)+ALKEFF
      GO TO 130
C
C     IF NO LIME IS ADDED (MLMPPH <= 0), THE LIQUID FLOW IS TOO LOW
C     AND THE LSD CAN NOT FUNCTION.  SET VARIABLE 'NOWAY' AND SKIP
C     CALCULATIONS.
C
  120 MLMPPH = LMPPH
      IF(MLMPPH .GT. 0.)
     &EFF1 = P(35) * (((1./(P(35)/(P(35)-ALKEFF)))) +
     &       ALLIMB/LMTEMP*75.)+ALKEFF
      IF (MLMPPH .GT. 0) GO TO 130
      EFF1= 75.+ALKEFF
C
C     CALCULATE NUMBER OF DRYERS.  SPARES ARE ASSUMED.
C
  130 EFFLSD= EFF1
      N = IFIX (BYPFAC * (ACFM/550000.)) + 1
      IF (N .LE. 4) MODNUM = N + 1
      IF ((N.GT.4) .AND. (N.LE.8)) MODNUM = N + 2
      IF (N .GE. 8) MODNUM = N + 3
C
C     CALCULATE L/G RATIO.
C
      LG = (WATPPH*2.) / ACFM
C
C     SAVE EFFICIENCY IN PARAMETER ARRAY (PARAMETER 35) FOR USE IN
C     OPTIMIZATION.
C
      P(35) = EFF1
C
C     DEBUG OUTPUT.
C
      WRITE (IODB,8100) NOWAY,EFF1,EFFLSD,BYPFAC,LMPPH,MLMPPH,WATPPH,
     &                  PCTSLD,ALKEFF,MODNUM,LG
 8100 FORMAT (T10,'NOWAY,EFF1,EFFLSD,BYPFAC,LMPPH,MLMPPH=',I1,5F12.4,
     &       /T10,'WATPPH,PCTSLD,ALKEFF,MODNUM,LG=',3F12.4,I2,F12.4)
C
C*************************************************************************
C
C     PART 2:  MODULE SPECIFIC OUTPUT.
C
C     THIS PART WRITES THE MODULE SPECIFIC OUTPUT FOR THE LSD.  THIS
C     INCLUDES THE MESSAGES AS TO LSD INOPERABILITY UNDER DEFINED
C     CONDITIONS.  NOTE, OUTPUT TO BOTH SCREEN AND HARD-COPY.
C
C     WRITE HEADER.
C
  200 IF (KO(1)) WRITE (IO6,6200)
      IF (KO(2)) WRITE (IO7,6200)
```

```
 6200 FORMAT (///T5,'LIME SPRAY DRYING')
C
C     TRANSFER TO PROPER OUTPUT BASED ON VARIABLE 'NOWAY'.
C
      N = NOWAY + 1
      GO TO (210,220,210,225),N
C
C     WRITE MODULE SPECIFIC OUTPUT FOR A COST CALCULABLE CASE (NOWAY=
C     0,2).
C
  210 IF (KO(1)) WRITE (IO6,6210) P(31),P(40),EFFLSD,MODNUM,LG,P(41)
     &       ,BYPFAC
      IF (KO(2)) WRITE (IO7,6210) P(31),P(40),EFFLSD,MODNUM,LG,P(41)
     &       ,BYPFAC
 6210 FORMAT (T10,'OVERALL STOICHIOMETRIC RATIO',T48,F5.2,
     &        /T10,'SPRAY-DOWN TEMPERATURE',T46,F5.0,' DEG F',
     &        /T10,'OVERALL SO2 REMOVAL EFFICIENCY',T47,F5.1,' %',
     &        /T10,'NUMBER OF DRYER UNITS',T47,I3,
     &        /T10,'DESIGN L/G',T45,F6.3,' GAL/1000 CU FT',
     &        /T10,'AVERAGE PRESSURE DROP',T48,F3.0,' IN H20',
     &        /T10,'FLUE GAS TREATED',T45,2PF6.2,'%')
      IF (NOWAY .EQ. 2) GO TO 230
      GO TO 300
C
C     WRITE MODULE SPECIFIC OUTPUT FOR LOW TEMPERATURE CASE (NOWAY=1).
C
  220 IF (KO(1)) WRITE (IO6,6220)
      IF (KO(2)) WRITE (IO7,6220)
 6220 FORMAT (T10,'** THIS MODULE WOULD NOT FUNCTION BECAUSE THE ',
     &             'TEMPERATURE IS NEAR',
     &        /T13,'OR BELOW SATURATION.')
      GO TO 700
  225 IF(KO(1)) WRITE(IO6,6225) P(36)
      IF(KO(2)) WRITE(IO7,6225) P(36)
 6225 FORMAT(T10,'** (LSD) THE PERCENT SOLIDS EXCEEDS ',F5.2,' --',
     &        /T10,'MORE THAN IS ALLOWED BY CURRENT DESIGN STANDARDS.',
     &      /T10,'NO FURTHER CALCULATIONS WILL BE MADE FOR THIS MODULE.')
      GO TO 700
C
C     WRITE MODULE SPECIFIC OUTPUT FOR LOW LIQUID RATE CASE (NOWAY=2).
C
  230 IF (KO(1)) WRITE (IO6,6230)
      IF (KO(2)) WRITE (IO7,6230)
 6230 FORMAT (T10,'** THIS MODULE WOULD NOT FUNCTION PROPERLY',
     &             'BECAUSE LIQUID RATE',
     &        /T13,'IS TOO LOW.')
      GO TO 300
C
C*************************************************************************
C
C     PART 3:  DIRECT CAPITAL COST CALCULATIONS.
C
```

```
C     THIS PART CALCULATES DIRECT CAPITAL COSTS AND STORES THEM IN
C     THE ARRAY 'CAPCST' FOR LATER OUTPUT.  NOTE THAT VERY DETAILED
C     COSTS ARE DONE FOR THE LSD MODULE BUT ONLY SUBTOTALS ARE
C     REPORTED.
C
C     CALCULATE COST OF 30 DAY STORAGE SILOS (CC(1)) AND CONVEYORS
C     (CC(2)). N IS THE NUMBER OF SILOS AND Q IS THE CAPACITY OF
C     EACH.
C
  300 N = IFIX (MLMPPH*0.000144) + 1
      Q = MLMPPH * 0.36 / N
      CC(1) = N * 265300. * (Q/2500.)**.68 * P(291)
      CC(2) = 86600. * (MLMPPH/2600.)**.81 * P(291)
      IF (N .GT. 1) CC(2) = CC(2) * 1.2**(N-1)
C
C     CALCULATE COST OF DAY BINS (CC(3)) AND CONVEYORS (CC(4)).  N IS
C     THE NUMBER OF BINS AND Q IS THE CAPACITY OF EACH.
C
      N = IFIX (MLMPPH/13000.) + 1
      Q = MLMPPH * 0.012 / N
      CC(3) = N * 36350. * (Q/156.)**.68 * P(291)
      CC(4) = 2590. * (MLMPPH/6667.)**.81 * P(291)
      IF (N .GT. 1) CC(4) = CC(4) * 1.2**(N-1)
C
C     CALCULATE COST OF BALL MILL (CC(5)) AND SCREW CONVEYOR (CC(6)).
C     N IS THE NUMBER OF BINS FROM ABOVE.
C
      CC(5) = 50516. * (MLMPPH/1667.)**.526 * 2. * P(291)
      CC(6) = 3540. * (MLMPPH/5000.)**.8 * N * P(291)
C
C     CALCULATE COST OF CLARIFIER (CC(7)).
C
      CC(7) = 97900. * (MLMPPH/13333.)**.65 * 2. * P(291)
C
C     CALCULATE COST OF PRODUCT TANK AND AGITATORS (CC(8)) AND PRO-
C     DUCT PUMPS (CC(9)).  N IS THE NUMBER OF TANKS AND Q IS THE CAP-
C     ACITY OF EACH.
C
      N = IFIX (MLMPPH/13151.938) + 1
      Q = MLMPPH * 0.8325 / N
      CCT = N * 56600. * (Q/10949.)**.68
      CCA = N * 13200.
      IF (Q .GT. 4500.) CCA = CCA * 2.
      CC(8) = (CCT + CCA) * P(291)
      CC(9) = 14400. * (MLMPPH/687000.)**.47 * 2. * P(291)
C
C     CALCULATE COST OF FEED TANK AND AGITATORS (CC(10)) AND FEED
C     PUMPS (CC(11)).  N IS THE NUMBER OF TANKS AND Q IS THE CAPA-
C     CITY OF EACH.
C
      N = IFIX ((PPHPRT/2.681 + MLMPPH/1.537) / 10949.) + 1
      Q = (PPHPRT/2.681 + MLMPPH/1.537) / N
```

```
      CCT = N * 56600. * (Q/10949.)**.68
      CCA = N * 13200.
      IF (N .GT. 4500.) CCA = CCA * 2.
      CC(10) = (CCT + CCA) * P(291)
      CC (11) = 14400. * (PPHPRT/537965. + MLMPPH/308465.)**.47 * 2. *
     & P(291)
C
C     CALCULATE COST OF DRYERS AND DUCTING (CC(12)).  N IS THE NUMBER
C     OF DRYERS CALCULATED EARLIER AS MODNUM AND Q IS THE CAPACITY OF
C     EACH.
C
      N = MODNUM
      Q = BYPFAC * ACFM/N
      CC(12) = N * 247. * Q**.7 * P(291)
C
C     CALCULATE THE COST OF CONSTANT HEAD TANKS AND AGITATORS (CC(13)).
C     N IS THE NUMBER OF DRYERS ABOVE AND Q IS THE CAPACITY OF EACH
C     TANK.
C
      Q = PPHPRT/10.665 + MLMPPH/6.115
      CCT = N * 56600. * (Q/10949.)**.68
      CCA = N * 13200.
      IF (Q .GT. 4500.) CCA = CCA * 2.
      CC(13) = (CCT + CCA) * P(291)
C
C     CALCULATE COST OF CONVEYOR (CC(14)) AND DRYER BOTTOMS SILO
C     (CC(15)).  N IS THE NUMBER OF DRYERS ABOVE.
C
      CC(14) = 2590. * ((PPHPRT+MLMPPH)/50000.)**.81 * N/2 * P(291)
      CC(15) = 265300. * ((PPHPRT+MLMPPH)/44620.)**.68 * P(291)
C
C     CALCULATE COST OF RECYCLE SOLIDS CONVEYOR (CC(16)).
C
      CC(16) = 86600. * ((PPHPRT+MLMPPH)/53190.)**.81 * P(291)
C
C     CALCULATE COST OF RECYCLE SLURRY TANKS AND AGITATORS (CC(17))
C     AND RECYCLE SLURRY PUMPS (CC(18)).  N IS THE NUMBER OF TANKS
C     AND Q IS THE CAPACITY OF EACH.
C
      N = IFIX ((PPHPRT+MLMPPH)/9784.2126) + 1
      Q = (PPHPRT+MLMPPH) * 1.1190 / N
      CCT = N * 56600. * (Q/10949.)**.68
      CCA = N * 13200.
      IF (Q .GT. 4500.) CCA = CCA * 2.
      CC(17) = (CCT + CCA) * P(291)
      CC(18) = 14400. * ((PPHPRT+MLMPPH)/562152.)**.47 * 2. * P(291)
C
C     CALCULATE SUBTOTALS FOR 'REAGENT PREP', 'ABSORBERS & DUCTING',
C     'RECYCLE/WASTE DISPOSAL' AND 'TOTAL LSD COST' AND STORE FOR
C     USE IN THE CAPITAL COST REPORT.  INSTALLATION FACTORS (PARAM-
C     ETER 42) BE APPLIED TO THE FIRST THREE.
C
```

```
      CAPCST(7,2) = ((CC(1) + CC(2) + CC(3) + CC(4) + CC(5) + CC(6) +
     &               CC(7) + CC(8) + CC(9) + CC(10)) * P(42)) * RATPLT
      CAPCST(7,3) = (CC(11) + CC(12)+ CC(13)) * P(42) *RATPLT
      CAPCST(7,4) =(CC(14) + CC(15) + CC(16) + CC(17) + CC(18)) * P(42)
     &             * RATPLT
      CAPCST(7,1) = CAPCST(7,2) + CAPCST(7,3) + CAPCST(7,4)
C
C     DEBUG OUTPUT.
C
      WRITE (IODB,8300) CC,(CAPCST(7,I),I=1,4)
 8300 FORMAT (T10,'CC=',18F12.2,/T13,9F12.2,
     &       T10,'CAPCST(LSD)=',4F12.2)
C
C*************************************************************************
C
C     PART 4:  INDIRECT CAPITAL COST CALCULATIONS.
C
C     THE INDIRECT CAPITAL COSTS ARE GENERALLY A PERCENTAGE OF THE
C     DIRECT CAPITAL COST OR DIRECT PLUS INDIRECT CAPITAL COSTS.  THE
C     PERCENTAGES ARE RETAINED IN THE PARAMETER FILE (PARAMETER 43
C     THRU 50).  EACH INDIRECT COST FOR EACH MODULE IS SUMMED INTO
C     THE ARRAY 'INDCST'.
C
C     CALCULATE ENGINEERING DESIGN AND SUPERVISION(TVA) OR
C     GENERAL FACILITIES(EPRI)
C
      IC(1) = P(43)/100. * CAPCST(7,1)
      INDCST(1) = INDCST(1) + IC(1)
C
C     CALCULATE A-E CONTRACTOR(TVA) OR ENG. AND HOME OFF. FEES (EPRI)
C
      IC(2) = P(44)/100. * CAPCST(7,1)
      INDCST(2) = INDCST(2) + IC(2)
C
C     CALCULATE CONSTRUCTION EXPENSE (TVA) PROJECT CONTINGENCY (EPRI)
C
      IC(3) = P(45)/100. * CAPCST(7,1)
      INDCST(3) = INDCST(3) + IC(3)
C
C     CALCULATE CONTRACTOR FEES (TVA) OR PROCESS CONTINGENCY (EPRI)
C
      IC(4) = P(46)/100. * CAPCST(7,1)
      INDCST(4) = INDCST(4) + IC(4)
C
C     SUM DIRECT AND INDIRECT CAPITAL COSTS TO HAVE A BASE FOR THE
C     REMAINDER OF THE INDIRECT COSTS.
C
      COST = CAPCST(7,1) + IC(1) + IC(2) + IC(3) + IC(4)
C
C     CALCULATE CONTINGENCY (TVA) OR SALES TAX (EPRI)
C
      IF (TVA) THEN
```

```
            IC(5) = P(47)/100. * COST
         ELSE
            IC(5) = P(47)/100. * CAPCST(7,1)
         ENDIF
        INDCST(5) = INDCST(5) + IC(5)
C
C       INCLUDE IC(5) IN RUNNING COST BASE TOTAL
C
        COST = COST + IC(5)
C
C        CALCULATE ROYALTIES (TVA) OR ROYALTY ALLOWANCE (EPRI)
C
        IC(6) = P(48)/100. * COST
        INDCST(6) = INDCST(6) + IC(6)
C
C        CALCULATE INTEREST DURING CONSTRUCTION (TVA)
C
        IF (TVA) THEN
          IC(8) = P(60)/100. * COST
          INDCST(8) = INDCST(8) + IC(8)
        ENDIF
C
C         CALCULATE ALLOWANCE FOR STARTUP AND MODIFICATIONS (TVA) OR
C                   INITIAL CATALYST (EPRI)
C
        IC(9) = P(52)/100. * COST
        INDCST(9) = INDCST(9) + IC(9)
C
C         CALCULATE LAND (TVA AND EPRI)
C
        IF (TVA) THEN
           IC(10)=0
         ELSE
           IC(10) = P(181)
        ENDIF
        INDCST(10) = INDCST(10) + IC(10)
C
C       DEBUG OUTPUT.
C
        WRITE (IODB,8400) IC,INDCST
 8400 FORMAT (T10,'IC=',8F10.2,
     &        /T10,'INDCST=',8F10.2)
C
C*******************************************************************************
C
C       PART 5:  ANNUAL OPERATING COST CALCULATIONS.
C
C       ANNUAL OPERATING COSTS ARE REPORTED FOR THE ENTIRE SYSTEM (ALL
C       MODULES).  MOST PARAMETERS ARE KEPT BY QUANTITY.  ALL VALUES
C       ARE ON A PER YEAR BASIS.
C
        AC(1) = P(53)
```

```
      ANN(1) = ANN(1) + AC(1)
C
C     CALCULATE ELECTRICITY REQUIREMENTS IN KILOWATTHOURS.
C
      AC(4) = P(56) * BSIZE * 1000. * CF
      ANN(4) = ANN(4) + AC(4)
C
C     CALCULATE WATER REQUIREMENTS IN 1000 GALLONS.
C
      AC(5) = WATPPH/7572.7 * 8760. * CF
      ANN(5) = ANN(5) + AC(5)
C
C     NO REHEAT REQUIRED.
C
C     THE ONLY CHEMICAL REQUIRED IS LIME.  CALCULATE ITS REQUIREMENT
C     IN TONS.
C
      AC(8) = MLMPPH/1739. * 8760. * CF
      ANN(8) = ANN(8) + AC(8)
C
C     MAINTENANCE IS A PERCENTAGE OF EQUIPMENT OR DIRECT CAP-
C     ITAL COST.
C
      AC(10) = P(59)/100. * CAPCST(7,1)
      ANN(10) = ANN(10) + AC(10)
C
C     NO FUEL COST PREMIUM.
C
C     ONLY DRY WASTES ARE PRODUCED.  CALCULATE THIS DISPOSAL QUANTITY
C     IN TONS.
C
      AC(13) = .667 * (PPHPRT+MLMPPH) * 8760. * CF / 2000.
      ANN(13) = ANN(13) + AC(13)
C
C     DEBUG OUTPUT.
C
      WRITE (IODB,8500) AC,ANN
 8500 FORMAT (T10,'AC=',7F10.2,
     &       /T13,6F10.2,
     &       /T10,'ANN=',7F10.2,
     &       /T14,6F10.2
     &       /T14,6F10.2
     &       /T14,6F10.2)
C
C*************************************************************************
C
C     PART 6:  MATERIAL BALANCE CALCULATIONS.
C
C     AT THE END OF EACH MODULE, THE STREAM PARAMETERS CONTAINED IN
C     THE COMMON 'STREAM' ARE ADJUSTED FOR THE AFFECTS OF THIS MOD-
C     ULE.
C
```

```
C      CALCULATE THE NEW FLUE GAS TEMPERATURE.
C
       TEMP = ((1.-BYPFAC) * (TEMP+460.)) + (BYPFAC * (P(40)+460.)) -
      &        460.
C
C      SAVE OLD FLUE GAS SO2 AND CALCULATE NEW.
C
       OPHSO2 = PPHSO2
       PPHSO2 = PPHSO2*(((1.-EFFLSD/100.)*BYPFAC)+(1.-BYPFAC))
C
C      SAVE OLD FLUE GAS PARTICULATES AND CALCULATE NEW.
C
       OPHPRT = PPHPRT
       PPHPRT = P(39)*(MLMPPH+BYPFAC*OPHPRT) + (1.-BYPFAC)*OPHPRT
C
C      CALCULATE NEW FLUE GAS ACFM.
C
       CACFM = (BYPFAC * ACFM * (620./(460.+TEMP))) + ((1.-BYPFAC) *
      &        ACFM)
       ACFM = CACFM + ((WATPPH/1080.) * (359.*(TEMP+460.)/492.))
C
C      CALCULATE NEW FLUE GAS FLOW (LB/HR).
C
       PPH = PPH + WATPPH - (OPHSO2-PPHSO2) - (OPHPRT-PPHPRT)
C
C      CALCULATE NEW FLUE GAS MOISTURE.
C
       PPHMST = PPHMST + WATPPH
C
C      CALCULATE NEW FLUE GAS ALKALINITY.
C
       CAOEFF= .75*ALLIMB/LMTEMP
       IF(CAOEFF .GT. .75) CAOEFF= .75
       PPHALK = BYPFAC*(PPHALK-(RECALK*(1.-P(32)/100.)) + (MLMPPH*
      &   (1.-P(35)/100.)) - (ALLIMB*(1.-CAOEFF))) + (1.-BYPFAC)*PPHALK
C
C      NO CHANGE TO FLUE GAS NOX.
C
C      ADJUST OVERALL PRESSURE DROP DUE TO THIS MODULE.
C
       DELTAP = DELTAP + P(41)*BYPFAC
C
C      DEBUG OUTPUT.
C
       WRITE (IODB,8600) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,
      &                  PPHNOX
 8600 FORMAT (T10,'PPH,ACFM,TEMP,PPHMST=',4F14.4,
      &       /T10,'PPHALK,PPHPRT,PPHSO2,PPHNOX=',4F14.4)
C
C*****************************************************************
C
C      PART 7:  CONCLUSION
```

```
C
  700 WRITE (IODB,8700)
 8700 FORMAT (T2,'LEAVING SUBROUTINE LSD.')
C
      RETURN
      END
```

```
C
C           INTEGRATED AIR POLLUTION CONTROL SYSTEM PROGRAM
C           ========== === ========= ======= ====== =======
C
C                         FABRIC FILTER ROUTINE
C                         ------ ------ -------
C
C
C
C
C       WRITTEN BY:  J. R. SCHOETTELKOTTE
C                    PEDCO COMPUTER SERVICES
C                    JUNE, 1983
C
C       WRITTEN FOR:  G. E. BAKER
C                     PEI ASSOC., INC.
C                     PEI PN 3568-3
C                     J. O. MILLIKEN
C                     U.S. EPA; IERL-RTP
C
C
C
C           THIS ROUTINE CALCULATES DESIGN PARAMETERS, CAPITAL AND
C       ANNUAL OPERATING COSTS, AND MATERIAL BALANCE FOR FABRIC FILTERS.
C       THE ALGORITHMS WERE DEVELOPED BY J. P. ABRAHAM AND Y. M. SHAH
C       AND REVIEWED BY G. E. BAKER ALL OF PEI ASSOC., INC.
C       THIS ROUTINE CALLS THE SUBROUTINE 'FIXATE' FOR COMPLETION OF
C       DRY SORBENT INJECTION COSTS ONLY WHEN APPLICABLE.
C
C
C
      SUBROUTINE FF
      LOGICAL    BYPASS, KO, TVA
      INTEGER    BOTTOM, CLNLVL, CT, FMB, TYPCT
      REAL       IC, INDCST, NBYPFF
      DIMENSION AC(13), ANN(25), CAPCST(10,8), CC(9), IAPCS(9), IC(6),
     &          INDCST(10), KO(2), P(500), TABLE(8,8)
      COMMON /COAL1/    HHV, PS, PA, CST, PNA, ALK, CL, FEO, PM
      COMMON /ESC/ IYC, RATPLT, RATLA2, RATMA2, RATOM
      COMMON /COSTS/  CAPCST, ANN, INDCST
      COMMON /FAN/    DELTAP, CSTFAN
      COMMON /INPUT/   BSIZE, CF, AHFX, AHXT, HHVROM, PSROM, PAROM, CSTROM,
     &                 PNAROM, ALKROM, CLROM, FEOROM, PMROM, HHVCLN, PSCLN,
     &                 PACLM, CSTCLN, PNACLN, ALKCLN, CLCLN, FEOCLN, PMCLN,
     &                 FMB, NR, IS, CT, TYPCT, CLNLVL, BOTTOM, NM, IAPCS, KO,
     &                 TVA
      COMMON /INSYSP/ CCEFF, UTEFF, BNHR, FANLOS, QIN, ALKIN
      COMMON /MATBAL/ TABLE
      COMMON /OPT/    BYPFGD, BYPFF, EFFESP, EFFLSD, EFFLIM
      COMMON /PARM/   P
      COMMON /STREAM/ PPH, ACFM, TEMP, PPHMST, PPHALK, PPHPRT, PPHSO2, PPHNOX
C
C
```

```
      BYPASS = .FALSE.
      IODB = 8
      IO6 = 6
      IO7 = 7
      DO 10 I=1,13
      AC(I) = 0.0
   10 CONTINUE
      DO 15 I=1.8
      CC(I) = 0.0
   15 CONTINUE
      DO 20 I=1,6
      IC(I) = 0.0
   20 CONTINUE
C
      WRITE (IODB,8000)
 8000 FORMAT (T2,'ENTERING SUBROUTINE FF')
      WRITE (IODB,8010) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,
     &                  PPHNOX,BYPFF,BSIZE,CF,(CAPCST (6,I),I=1,10),
     &                  INDCST,ANN,(P(J),J=71,90)
 8010 FORMAT (T10,'PPH,ACFM,TEMP,PPHMST,PPHALK=',5F14.4,
     &       /T10,'PPHPRT,PPHSO2,PPHNOX,BYPFF,BSIZE,CF=',6F14.4,
     &       /T10,'CAPCST(FF)=',10F10.2,
     &       /T10,'INDCST=',8F10.2,
     &       /T10,'ANN=',7F10.2,
     &       /T14,6F10.2,
     &       /T14,6F10.2,
     &       /T14,6F10.2,
     &       /T10,'P(71-90)=',10F10.2,
     &       /T19,10F10.2)
C
C*******************************************************************************
C
C     PART 1:  SIZING CALCULATIONS.
C
C     THIS PART CALCULATES SIZING PARAMETERS FOR USE LATER IN THE
C     PROGRAM.
C
C     IF THERE IS BYPASS SET THE BYPASS SWITCH.
C
      IF (BYPFF .NE. 0.) BYPASS = .TRUE.
C
C     CALCULATE FRACTION NOT BYPASSED.
C
      NBYPFF = 1. - BYPFF
C
C     SPLIT INCOMING GAS STREAM (ACFM) TO BYPASS AND TREATED.
C
      BYACFM = ACFM * BYPFF
      TRACFM = ACFM - BYACFM
C
C     CALCULATE THE ASH RATE IN TONS PER HOUR.
C
```

```
      ASHRT = P(72)/100. * PPHPRT * NBYPFF / 2000.
C
C     DEBUG OUTPUT.
C
      WRITE (IODB,8100) NBYPFF,BYACFM,TRACFM,ASHRT
 8100 FORMAT (T10,'NBYPFF,BYACFM,TRACFM,ASHRT=',4F14.4)
C
C*****************************************************************************
C
C     PART 2:  MODULE SPECIFIC OUTPUT.
C
C     THIS PART WRITES THE MODULE SPECIFIC OUTPUT FOR THE FABRIC
C     FILTER.
C
C     WRITE HEADER.
C
      IF (KO(1)) WRITE (IO6,6200)
      IF (KO(2)) WRITE (IO7,6200)
 6200 FORMAT (///T5,'FABRIC FILTER')
C
C     WRITE DETAIL MODULE-SPECIFIC OUTPUT.
C
      IF (KO(1)) WRITE (IO6,6210) P(72),P(71),BYPFF
      IF (KO(2)) WRITE (IO7,6210) P(72),P(71),BYPFF
 6210 FORMAT (T10,'THE FABRIC FILTER IS DESIGNED TO REMOVE ',F4.1,
     &            '% OF THE PARTICULATE',
     &       /T10,'LOADING WITH AN AIR-TO-CLOTH RATIO OF ',F3.1,'. ',
     &            F4.1,'% OF THE FLUE GAS',
     &       /T10,'IS BYPASSING THE FABRIC FILTER.  THE FABRIC ',
     &            'FILTER REFLECTS A',
     &       /T10,'REVERSE AIR CLEANING CONFIGURATION AND TEFLON-',
     &            'COATED FIBERGLASS',
     &       /T10,'BAGS.')
C
C*****************************************************************************
C
C     PART 3:  DIRECT CAPITAL COST CALCULATION.
C
C     THIS PART CALCULATES THE DIRECT CAPITAL COSTS FOR THE FABRIC
C     FILTER.  THESE INCLUDE THE FILTER ITSELF, THE DUCTING, AND
C     ASH HANDLING SYSTEM.  AN INSTALLATION FACTOR (PARAMETER 74)
C     IS INCLUDED WHERE NEEDED.
C
C     CALCULATE THE COST OF THE FABRIC FILTER (CC(1)).  IT IS BASED
C     ON ACFM TREATED AND THE AIR-TO-CLOTH RATIO (PARAMETER 71).
C
      CC(1) = (215416. + 7.29*TRACFM/P(71)) * P(74)
C
C     CALCULATE THE COST OF DUCTING (CC(2)).  NOTE, TWO KINDS OF
C     DUCTING ARE USED:  THE FIRST IS FOR A RANGE OF 250 TO 649
C     MW AND THE OTHER FOR A RANGE OF 650 TO 1300 MW.
C
```

```
      IF (BSIZE .LT. 650.) CC(2) = (336.*TRACFM**.5 - 1575.) * P(74)
      IF (BSIZE .GE. 650.) CC(2) = (0.0522*TRACFM + 693.*TRACFM**.5
     &                              - 3399.) * P(74)
C
C     CALCULATE THE COST OF BYPASS DUCTING (CC(3)) IF A BYPASS IS
C     USED.
C
      IF (.NOT. BYPASS) GO TO 300
      CC(3) = (895.*BYACFM**.5 - 4004.64) * P(74)
C
C     CALCULATE THE COST OF ASH HOPPERS (CC(4)).
C
  300 CC(4) = 197.6 * BSIZE
C
C     CALCULATE THE COST OF PNEUMATIC PIPELINE (CC(5)).
C
      CC(5) = 22480. * ASHRT**.36
C
C     CALCULATE COST OF VACUUM (CC(6)).
C
      CC(6) = 50310. * ASHRT**.279
C
C     CALCULATE COST OF SILO BAGHOUSE (CC(7)).
      N = IFIX(ASHRT/100.) + 1
      CC(7) = N * 29620. * ASHRT**.662
C
C     CALCULATE COST OF SILOS (CC(8)).  NOTE N IS THE NUMBER OF SILOS
C     REQUIRED.
C
C
      CC(8) = N * 503100. * ASHRT**.279
C
C     CALCULATE REVERSE AIR FAN COST.
C
      CC(9)= 2.02*(.0624*ACFM)
C
C     CALCULATE SUBTOTALS FOR 'FABRIC FILTER & DUCTING', 'ASH HAND-
C     LING', AND 'TOTAL FF COST'.
C
      CAPCST(6,2) = (CC(1) + CC(2) + CC(3) + CC(9)) * RATPLT *P(293)
      CAPCST(6,3) = (CC(4) + CC(5) + CC(6) + CC(7) + CC(8)) *RATPLT *
     & P(293)
      CAPCST(6,1) = CAPCST(6,2) + CAPCST(6,3) * P(293)
C
C     DEBUG OUTPUT.
C
      WRITE (IODB,8300) CC,(CAPCST(6,I),I=1,3)
 8300 FORMAT (T10,'CC=',9F12.2,
     &       /T10,'CAPCST(FF)=',3F12.2)
C
C*******************************************************************************
C
```

```
C      PART 4:  INDIRECT CAPITAL COST CALCULATIONS.
C
C      THE INDIRECT CAPITAL COSTS ARE GENERALLY A PERCENTAGE OF THE
C      DIRECT CAPITAL COST OR DIRECT PLUS INDIRECT CAPITAL COSTS.  THE
C      PERCENTAGES ARE RETAINED IN THE PARAMETER FILE (PARAMETERS 75
C      THRU 80).  EACH INDIRECT COST FOR EACH MODULE IS SUMMED INTO
C      THE ARRAY 'INDCST'.
C
C       CALCULATE ENGINEERING DESIGN AND SUPERVISION (TVA) OR
C       GENERAL FACILITIES (EPRI)
C
       IC(1) = P(76)/100. * CAPCST(6,1)
       INDCST(1) = INDCST(1) + IC(1)
C
C       CALCULATE A-E CONTRACTOR(TVA) OR ENG. AND HOME OFF. FEES (EPRI)
C
       IC(2) = P(75)/100. * CAPCST(6,1)
       INDCST(2) = INDCST(2) + IC(2)
C
C       CALCULATE CONSTRUCTION EXPENSE (TVA) PROJECT CONTINGENCY (EPRI)
C
       IC(3) = P(77)/100. * CAPCST(6,1)
       INDCST(3) = INDCST(3) + IC(3)
C
C       CALCULATE CONTRACTOR FEES (TVA) OR PROCESS CONTINGENCY (EPRI)
C
       IC(4) = P(78)/100. * CAPCST(6,1)
       INDCST(4) = INDCST(4) + IC(4)
C
C       SUM DIRECT AND INDIRECT CAPITAL COSTS TO HAVE A BASE FOR THE
C       REMAINDER OF THE INDIRECT COSTS.
C
       COST = CAPCST(6,1) + IC(1) + IC(2) + IC(3) + IC(4)
C
C       CALCULATE CONTINGENCY (TVA) OR SALES TAX (EPRI)
C
       IF (TVA) THEN
             IC(5) = P(79)/100. * COST
          ELSE
             IC(5) = P(79)/100. * CAPCST(6,1)
          ENDIF
       INDCST(5) = INDCST(5) + IC(5)
C
C      INCLUDE IC(5) IN RUNNING COST BASE TOTAL
C
       COST = COST + IC(5)
C
C       CALCULATE ROYALTIES (TVA) OR ROYALTY ALLOWANCE (EPRI)
C
       IC(6) = P(80)/100. * COST
       INDCST(6) = INDCST(6) + IC(6)
C
```

```
C       CALCULATE INTEREST DURING CONSTRUCTION (TVA)
C
        IF (TVA) THEN
          IC(8) = P(142)/100. * COST
          INDCST(8) = INDCST(8) + IC(8)
        ENDIF
C
C         CALCULATE ALLOWANCE FOR STARTUP AND MODIFICATIONS (TVA) OR
C                   INITIAL CATALYST (EPRI)
C
        IC(9) = P(81)/100. * COST
        INDCST(9) = INDCST(9) + IC(9)
C
C         CALCULATE LAND (TVA AND EPRI)
C
        IC(10) = P(181)
        INDCST(10) = INDCST(10) + IC(10)
C
C       DEBUG OUTPUT.
C
        WRITE (IODB,8400) IC,INDCST
 8400 FORMAT (T10,'IC=',6F12.2,
     &       /T10,'INDCST=',6F12.2)
C
C*******************************************************************************
C
C       PART 5:  ANNUAL OPERATING COST CALCULATIONS.
C
C       ANNUAL OPERATING COSTS ARE REPORTED FOR THE ENTIRE SYSTEM (ALL
C       MODULES).  MOST PARAMETERS ARE KEPT BY QUANTITY.  ALL VALUES
C       ARE ON A PER YEAR BASIS.
C
C       THERE ARE 3 KINDS OF LABOR:  OPERATING, SUPERVISION, AND MAIN-
C       TENANCE.  THE OPERATIONAL LABOR IS A FUNCTION OF GAS FLOW.
C       SUPERVISORY LABOR IS A PERCENTAGE (PARAMETER 82) OF THE OPERAT-
C       ING LABOR.
C
        AC(1) = (3.51*TRACFM**.4 + .00146*TRACFM)*(1.+P(82)*.01)
        ANN(1) = ANN(1) + AC(1)
C
C       CALCULATE ELECTRICITY REQUIREMENT IN KILOWATTHOURS.
C
        HP = 113.99 * ALOG(ASHRT*2000.) - 1084.7
       &+.1*ACFM*DELTAP/(6356.*.9*.8)
        IF(HP .LT. 50.) HP = 50.
        AC(4) = .746*HP * CF * 8760.
        ANN(4) = ANN(4) + AC(4)
C
C       CALCULATE WATER REQUIREMENT IN 1000 GALLONS.  NOTE, A DISPOSAL
C       FRACTION 'DISFCT' IS DETERMINED BASED ON THE USE OF A LIME
C       SPRAY DRYER.  PARAMETER 83 IS THE PROPORTION OF WATER ADDED TO
C       THE ASH.
```

```
C
C     IF LIMESTONE FGD W/ CHEMICAL FIXATION IS USED, DISFCT=0.
      DISFCT = 1.0
      DO 500 I=1,9
      IF (IAPCS(I) .EQ. 7) DISFCT = .334
      IF(IAPCS(I) .EQ. 8 .AND. P(230) .EQ. 2.) DISFCT= 0.
  500 CONTINUE
      AC(5) = 2100.72 * DISFCT * ASHRT * CF * (P(83)/100.)
      ANN(5) = ANN(5) + AC(5)
C
C     NO REHEAT IS REQUIRED.
C
C     NO CHEMICALS ARE REQUIRED.
C
C     MAINTENANCE INCLUDE REBAGGING IN ADDITION TO REGULAR
C     MAINTENANCE.
C
      RBCOST = 0.346 * TRACFM/P(71)
      AC(10) = RBCOST + CAPCST(6,1)*P(90)*.01
      ANN(10) = ANN(10) + AC(10)
C
C     NO FUEL COST PREMIUMS.
C
C     ONLY DRY WASTES ARE PRODUCED.  CALCULATE THIS DISPOSAL QUANTITY
C     IN TONS.
C
      AC(13) = DISFCT * (1.+P(83)/100.) * ASHRT * 8760. * CF
      ANN(13) = ANN(13) + AC(13)
C
C     DEBUG OUTPUT.
C
      WRITE (IODB,8500) AC,ANN
 8500 FORMAT (T10,'AC=',7F10.2,
     &        /T13,6F10.2,
     &        /T10,'ANN=',7F10.2,
     &        /T14,6F10.2,
     &        /T14,6F10.2,
     &        /T14,6F10.2)
C
C*****************************************************************
C
C     PART 6:  MATERIAL BALANCE CALCULATIONS.
C
C     AT THE END OF EACH MODULE, THE STREAM PARAMETERS CONTAINED IN
C     THE COMMON 'STREAM' ARE ADJUSTED FOR THE AFFECTS OF THIS MODULE.
C
C     CALCULATE NEW FLUE GAS ALKALINITY.
C
      PPHALK = PPHALK - (P(72)/100.)*NBYPFF*PPHALK
C
C     CALCULATE NEW FLUE GAS PARTICULATES.
C
```

```
      OPHPRT= PPHPRT
      PPHPRT = PPHPRT - ASHRT*2000.
C
C     CALCULATE NEW FLUE GAS SO2.  FIRST, DETERMINE SO2 EFFICIENCY
C     BASED ON PRECEEDING MODULES.
C
      SO2EFF = 0.0
      DO 600 I=1,9
      IF (IAPCS(I) .EQ. 2) SO2EFT = P(85)/100.
      IF (IAPCS(I) .EQ. 4) SO2EFT = P(86)/100.
      IF (IAPCS(I) .EQ. 7) SO2EFT = P(87)/100.
      IF (IAPCS(I) .EQ. 9) SO2EFT = P(88)/100.
      IF (SO2EFT .GT. SO2EFF) SO2EFF = SO2EFT
  600 CONTINUE
      OPHSO2= PPHSO2
      PPHSO2 = OPHSO2*(1.-SO2EFF)*NBYPFF + OPHSO2*BYPFF
C
C     CALCULATE NEW FLUE GAS NOX.  NOX ONLY CHANGES IF DRY SORBENT
C     INJECTION IS USED.
CC
      OPHNOX = PPHNOX
      DO 610 I=1,9
      IF (IAPCS(I) .EQ. 9) PPHNOX = OPHNOX*NBYPFF*.88
     &                            + OPHNOX*BYPFF
  610 CONTINUE
C
C     CALCULATE NEW FLUE GAS FLOW (LBS/HR)
C
      PPH = PPH - ASHRT*2000. - (OPHSO2-PPHSO2) -
     &      (OPHNOX-PPHNOX)
C
C     NO CHANGE TO FLUE GAS ACFM, TEMPERATURE, OR MOISTURE.
C
C     ADJUST OVERALL PRESSURE DROP DUE TO THIS MODULE.
C
      DELTAP = DELTAP + P(84)
C
C     DEBUG OUTPUT.
C
      WRITE (IODB,8600) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,
     &                  PPHNOX
 8600 FORMAT (T10,'PPH,ACFM,TEMP,PPHMST=',4F14.4,
     &       /T10,'PPHALK,PPHPRT,PPHSO2,PPHNOX=',4F14.4)
C
C*******************************************************************************
C
C*******************************************************************************
CC
C     PART 7: FIXATION CONSIDERATIONS.
C
C     IF A DRY SORBENT INJECTION MODULE PRECEEDS THIS FABRIC FILTER,
C     THE WASTES PRODUCED MUST BE FIXATED DUE TO NAHCOLITE CONTAMIN-
```

```
C     ATION. THIS PART ADJUSTS THE PREVIOUSLY CALCULATED WASTES AND
C     THEN CALLS THE SUBROUTINE 'FIXATE' TO COMPLETE THE DSI COSTS.
C
C     DETERMINE IF DSI HAS PRECEEDED THIS MODULE. IF NOT, SKIP TO END.
C
C
      DO 700 I=1,9
      IF(IAPCS(I) .EQ. 9) GO TO 710
  700 CONTINUE
      GO TO 800
C
C     REMOVE THE PREVIOUSLY CALCULATED DRY WASTES FROM THE ANNUAL
C     COSTING ARRAY.
C
  710 ANN(13)= ANN(13)-AC(13)
C
C     CALCULATE THE SO2 AND ASH IN TONS/HOUR  REMOVED BY THIS FABRIC FIL
C
      SO2= (OPHSO2-PPHSO2)/2000.
      ASH= (OPHPRT-PPHPRT)/2000.
C
C     CALL SUBROUTINE 'FIXATE' TO FINISH DSI COSTS.
C
      CALL FIXATE(SO2,ASH)
C
C     PART 8:  CONCLUSION
C
  800 WRITE (IODB,8800)
 8800 FORMAT (T2,'LEAVING SUBROUTINE FF.')
C
      RETURN
      END
```

```
$DEBUG
C
C
C          THIS SUBROUTINE CALCULATES THE SYSTEM'S CAPITAL AND ANNUAL
C      FAN COST (IF ANY). THE EQUATIONS ARE ADAPTED FROM PEDCO ENVIRON-
C      MENTAL'S "PISCES III" MODEL.
C
C          WRITTEN BY: PEDCO ENVIRONMENTAL INC. (JP)
C                      6/83
C
C
      SUBROUTINE FAN1
      INTEGER FMB,CT,BOTTOM,TYPCT,CLNLVL,R
      LOGICAL KO(2),TVA
      REAL INDCST,KWH,ML,IC,NTRAIN
      DIMENSION ELECT(10,7,2),RHP(10),COST(7)
      DIMENSION HMOD(9),HSTRM(5,8),NST(9),HCAP(10,9),SHCAP(6,3,9),AC(13)
      DIMENSION A(3),B(3),C(3),ML(3),HPCAT(2),IC(6)
      COMMON /COSTS/ CAPCST(10,8),ANN(25),INDCST(10)
      COMMON /STREAM/ PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
      COMMON /INPUT/BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,HHVCLN,PSCLN,PACLN,CSTCLN,PNACLN,ALKCLN,
     &CLCLN,FEOCLN,PMCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,IAPCS(9),KO
     &,TVA
      COMMON /ESC/ IYC,IMC,IYA,IMA,RATPLT,RATLAB,RATMAT,RATOM
      COMMON /PARM/ P(500)
      COMMON /FAN/ DELTAP,CSTFAN
      COMMON /TVAFAN/ TGAS,TWG,VPD,NTRAIN,DLTAPF,NOTRAN,NTOTAL,ISCRUB,
     &                NOREDN,PD,DPRH
      DATA A/360.23,82.29,135.0/,B/0.8425,1.0,1.0/,
     &     C/0.0,66071.,-162500./,ML/250.,500.,700./,HPCAT/800.,5000./
      DATA ELECT/355.,370.,524.,2*762.,2*1672.,2*3785.,6375.,
     &           0.20,0.27,0.42,6*0.89,0.00205602,
     &           0.92,0.96,1.02,6*1.10,0.00514005,
     &           0.72,0.86,1.16,6*1.86,0.0076956,
     &           7*1.,2.,2*3.,
     &           190.,300.,510.,900.,2*1790.,3*70.7362,471.027,
     &           5*186.311,3*0.27048,2*204.728,
     &           10*0.0,
     &           9*0.0,1.00048,
     &           9*0.0,1.00048,
     &           9*0.0,0.948743,
     &           10*0.0,
     &           6*0.0,3*0.89287,0.639446,
     &           5*0.33909,3*1.80974,2*0.855260/
      DATA RHP/2.5,7.5,15.,35.,50.,90.,133.,233.,250.,1500./
C
C*******************************************************************************
C
C     PART 1: SIZING CALCULATIONS.
C
C      DETERMINE IF FGD IS PRESENT IN THE SYSTEM
```

```
C
C      IFGD= 0: NO FGD IN SYSTEM
C          = 1: FGD PRESENT BUT NOT FINAL MODULE
C          = 2: FGD IS LAST MODULE
C
       PDV= 0.
       IFGD= 0
       DO 2 I=1,9
       IF (IAPCS(I) .NE. 8) GOTO 2
       IFGD=1
       IF(I .EQ. 9 .OR. IAPCS(I+1) .EQ.10) IFGD=2
     2 CONTINUE
       WRITE(8,*) ' IFGD=',IFGD
       IF(IFGD .NE. 0) GOTO 3
C      CALCULATE TOTAL FAN HORSEPOWER REQUIRED.
C
       HPTOT= ACFM*DELTAP*1.1/(6356.*0.9*0.8)
       GOTO 4
     3 TGASC = (TGAS + 460 + TWG) / (460 + TWG)
       EN = NTRAIN * 8.33 * 60.
       IF(ISCRUB.GT.2) PDV=VPD
       GASFLO = (ACFM + (EN*379)/(60.*18.016)) * TGASC
       CFMDUC = GASFLO / NOTRAN
       IF (DLTAPF.EQ.0.0) DLTAPF=(PD+PDV+DPRH+3.5)*1.15
       DELTAP= DELTAP+DLTAPF
       FANHP= CFMDUC*DELTAP/(6356.*.70)
       HPTOT= FANHP*NOTRAN
C
C      DETERMINE NUMBER OF FANS REQUIRED.
C
     4 IF(HPTOT .LE. 5000. .AND. ACFM .LE. 1.E6) NFANS= 1
       IF(HPTOT .GT. 5000. .AND. ACFM .LE. 1.E6) NFANS= 2
       IF(ACFM .GT. 1.E6) NFANS=3
       IF (IFGD .EQ. 2) NFANS= NTOTAL
C
C      CALCULATE HP PER FAN.
C
       IF(IFGD .NE. 0) GO TO 10
       FANHP= HPTOT/NFANS
       CFMDUC=ACFM/NFANS
C
C*******************************************************************************
C
C      PART 2: MODULE SPECIFIC OUTPUT.
C
C      LOOP ON OUTPUT DEVICE.
C
    10 DO 30 LD=6,7
    30 IF(KO(LD-5)) WRITE(LD,1000) DELTAP,NFANS,FANHP
  1000 FORMAT(///T5,'FANS',/T10,'THE TOTAL SYSTEM PRESSURE DROP IS',
      &       F5.1,' IN. H20.',/T10,'THE SYSTEM REQUIRES ',I1,' FAN(S)',
      &       ' RATED AT ',F6.0,' HP EACH. ')
```

```
C    > WRITE DEBUG OUTPUT <
C
C*****************************************************************************
C
C      PART 3: DIRECT CAPITAL COST CALCULATIONS.
C
C
      FANMC1=51.2686*CFMDUC**.61*DELTAP**.1738
      FANMC1=FANMC1*(2.57164 + DELTAP * 0.0693867)
      FANMC2=357.1*FANHP**.66224
      FANMC3=25700 + 1517.*DELTAP
      FANLC= (.052*FANMC1+.035*FANMC2+.02*FANMC3)*NFANS
      FANMC=NFANS*(FANMC1+FANMC2+FANMC3)
C
C      ELECTRICAL EQUIPMENT COST CALCULATIONS
C
      EHP=FANHP
      NOPMOT= NFANS
      NSPMOT= NOREDN
      INOMOT= NOPMOT+NSPMOT
      DO 1130 IN=1,10
      IF(EHP .LE. RHP(IN)) GO TO 1140
 1130 CONTINUE
      IN=10
 1140 II=IN
      DO 1160 JJ=1,5
      IF(EHP .EQ. 0.) GO TO 1150
      COST(JJ)= ELECT(II,JJ,1)*EHP**ELECT(II,JJ,2)
      GO TO 1160
 1150 COST(JJ)= 0.
 1160 CONTINUE
      STRM= COST(1)*INOMOT+STRM
      POWM= (COST(2)+COST(4))*INOMOT+POWM
      POWL= (COST(3)+10.*COST(4))*INOMOT+POWL
      IF(COST(5)-2.) 1170,1180,1190
 1170 HP480=EHP*INOMOT+HP480
      NSMOT=INOMOT+NSMOT
      GO TO 1200
 1180 NMMOT=INOMOT+NMMOT
      GO TO 1200
 1190 NLMOT= INOMOT+NLMOT
 1200 ARHP=EHP*INOMOT
      OPHP=EHP*NOPMOT
      SPHP=EHP*NSPMOT
 1210 CONTINUE
 1220 CONTINUE
C
C      "WIDE AREA" HORSE POWER REQUIREMENTS
C
      PSARHP=ARHP*10.
C
      REMCTR= 250.
```

```
      PSDIST= 500.
      AMOTDT= 250.
C ELECTRICAL MOTOR CONTROL COSTS
      DO1330 MA = 1,8
      NOTMOT=NSMOT+NMMOT+NLMOT
      NCARD = (NOTMOT-1)/20+1
      CTRMC=(NSMOT+NMMOT)*675.+NCARD*2.97*
     &REMCTR+NLMOT*(3340.+2.11*REMCTR+2.23*AMOTDT)
      CTRLC=(NSMOT+NMMOT)*81.+NCARD*21.35*
     &REMCTR+NLMOT*(147.+21.27*REMCTR+10.64*AMOTDT)
      WRITE(8,*) ' BEFORE ELECTRICAL MOTOR POWER ....'
C     ELECTRICAL MOTOR POWER COSTS
      POWMMC=STRM+NOTMOT*(480.+.80*AMOTDT)+
     &POWM*AMOTDT
      POWMLC=.25*STRM+NOTMOT*(80.+6.83*AMOTDT)+
     &POWL*AMOTDT
C     ELECTRICAL SUPPLY TO AREAS
      NCIRCB=2*IFIX((PSARHP-1.)/(2714.*5.)+1.)
      IF(NCIRCB.LT.1) NCIRCB=1
      NCABLE=(PSARHP-1)/2714+1
      IF(NCABLE.LT.1) NCABLE=1
      HPCABL=PSARHP/NCABLE
      IF(NCABLE.GT.8)GO TO 1280
      IF(PSARHP.GT.470.)GO TO 1230
      POWSMC=1.10*PSDIST+1.17*PSDIST+48600.
      POWSLC=2.75*PSDIST+11.70*PSDIST+3400.
      GO TO 1290
 1230 WIREMC=NCABLE*2.05060E-3*HPCABL**1.00048*PSDIST
      WIRELC=NCABLE*5.14005E-3*HPCABL**1.00048*PSDIST
      IF(PSARHP.GT.300.)GO TO 1240
      CONDMC=1.17*PSDIST
      CONDLC=11.70*PSDIST
      GO TO 1250
 1240 CONDMC=NCABLE*6.31298E-4*HPCABL**1.29600*PSDIST
      CONDLC=10.*CONDMC
      WRITE(8,*) ' BEFORE GO TO 1260...'
 1250 IF(PSARHP.GT.8150.)GO TO 1260
      CIRBMC=48600.
      CIRBLC=3400.
      GO TO 1270
 1260 CIRBMC=NCIRCB*29700.
      CIRBLC=NCIRCB*1700.
 1270 POWSMC=WIREMC+CONDMC+CIRBMC
      POWSLC=WIRELC+CONDLC+CIRBLC
      GO TO 1290
C     WIRE AND CONDUIT AT 2714 KVA
 1280 POWSMC=NCABLE*21.66*PSDIST+NCIRCB*29700.
      POWSLC=NCABLE*170.52*PSDIST+NCIRCB*1700.
C     AREA TRANSFORMER COSTS
 1290 IF(HP480.GT.10.)GO TO 1300
      TRFRMC=360.
      GO TO 1320
```

```
 1300 IF(HP480.GT.75.)GO TO 1310
      TRFRMC= 750.
      GO TO 1320
 1310 TRFRMC=111.977*HP480**.75169
 1320 TRFRLC= 125. + 0.10*TRFRMC
C     TOTAL ELECTRICAL AREA COSTS
      ELEMAT=(CTRMC+POWMMC+POWSMC+TRFRMC)
      ELELAB=(CTRLC+POWMLC+POWSLC+TRFRLC)
      WRITE(8,*) ' BEFORE 1880 CONTINUE...'
 1330 CONTINUE
C
C    >> TOTAL CAPITAL COST <<
C      (INCLUDE SERVICES AND MISCELLANEOUS,SALES TAX AND FREIGHT)
C
      CSTFAN=((FANMC+ELEMAT)*RATMAT+(ELELAB+FANLC)*RATLAB)*(1.+(P(375)+
     &       P(374))*.01)*(1.+P(376)*.01)
C
C*******************************************************************************
C
C     PART 4: INDIRECT CAPITAL COST CALCULATIONS.
C
C     INDIRECT COSTS ARE PERCENTAGES OF DIRECTS AND THEIR FACTORS ARE
C     CONTAINED IN THE PARAMETER FILE.
C
C
      CST= CSTFAN *P(289)
C
C      CALCULATE ENGINEERING DESIGN AND SUPERVISION (TVA) OR
C                GENERAL FACILITIES (EPRI)
C
      IC(1) = P(24)/100. * CST
      INDCST(1) = INDCST(1) + IC(1)
C
C      CALCULATE A-E CONTRACTOR(TVA) OR ENG. AND HOME OFF. FEES (EPRI)
C
      WRITE(8,*) ' BEFORE IC(2)...'
      IC(2) = P(23)/100. * CST
      INDCST(2) = INDCST(2) + IC(2)
C
C      CALCULATE CONSTRUCTION EXPENSE (TVA) PROJECT CONTINGENCY (EPRI)
C
      IC(3) = P(25)/100. * CST
      INDCST(3) = INDCST(3) + IC(3)
C
C      CALCULATE CONTRACTOR FEES (TVA) OR PROCESS CONTINGENCY (EPRI)
C
      IC(4) = P(26)/100. * CST
      INDCST(4) = INDCST(4) + IC(4)
C
C      SUM DIRECT AND INDIRECT CAPITAL COSTS TO HAVE A BASE FOR THE
C      REMAINDER OF THE INDIRECT COSTS.
C
```

```
      CST = CST + IC(1) + IC(2) + IC(3) + IC(4)
C
C      CALCULATE CONTINGENCY (TVA) OR SALES TAX (EPRI)
C
      IF (TVA) THEN
      IC(5) = P(27)/100. * CST
      ELSE
      IC(5)=  P(27)/100.*CSTFAN
      WRITE(8,*) ' AFTER IC(5...'
      ENDIF
      INDCST(5) = INDCST(5) + IC(5)
C
C     INCLUDE IC(5) IN RUNNING COST BASE TOTAL
C
      CST = CST + IC(5)
C
C      CALCULATE ROYALTIES (TVA) OR ROYALTY ALLOWANCE (EPRI)
C
      IC(6) = P(28)/100. * CST
      INDCST(6) = INDCST(6) + IC(6)
C
C      CALCULATE INTEREST DURING CONSTRUCTION (TVA)
C
      IC(8) = P(147)/100. * CST
      INDCST(8) = INDCST(8) + IC(8)
C
C       CALCULATE ALLOWANCE FOR STARTUP AND MODIFICATIONS (TVA) OR
C                 INITIAL CATALYST (EPRI)
C
      IC(9) = P(29)/100. * CST
      INDCST(9) = INDCST(9) + IC(9)
C
C       CALCULATE LAND (TVA AND EPRI)
C
      WRITE(8,*) ' BEFORE IC(10...'
      IC(10) = P(148)
      INDCST(10) = INDCST(10) + IC(10)
C
C*****************************************************************************
C
C     PART 5: ANNUAL OPERATING COST CALCULATIONS.
C
C     ELECTRICITY
C
      KWH= (.6*CF+.4)*HPTOT*.746*(530./(TEMP+460.))*8760.*CF
      ANN(4)= ANN(4)+KWH
C
C     MAINTENANCE LABOR AND MATERIALS
C
      ANN(10)= ANN(10)+P(30)*.01*CSTFAN
C
C*****************************************************************************
```

```
C
C     PART 6: MATERIAL BALANCE CALCULATIONS.
C
C     NONE.
C
C
C*****************************************************************************
C
C     PART 7: CONCLUSION.
C
C     WRITE DEBUG OUTPUT.
C
      WRITE(8,*) ' BEFORE RETURN...'
      RETURN
      END
```

```
*DEBUG
C
C           INTEGRATED AIR POLLUTION CONTROL SYSTEM PROGRAM
C           ========== === ========= ======= ====== =======
C
C                 ELECTROSTATIC PRECIPITATOR ROUTINE
C                 ----------------------------------
C
C
C      WRITTEN BY:    J.R. SCHOTTELKOTTE
C                     PEDCO COMPUTER SERVICES
C                     JUNE, 1983
C
C      WRITTEN FOR:   G.E. BAKER
C                         PEI ASSOC., INC.
C                         PEI PN 3568-3
C                     J.O. MILLIKEN
C                         US EPA;   IERL-RTP
C
C
C           THIS SUBROUTINE CALCULATES DESIGN PARAMETERS, CAPITAL AND
C      ANNUAL OPERATING COSTS, AND MATERIAL BALANCE FOR ELECTROSTATIC
C      PRECIPITATORS (ESP). THE ALGORITHMS WERE DEVELOPED BY J.P.
C      ABRAHAM AND Y.M. SHAH AND REVIEWED BY G.E. BAKER ALL OF PEI
C      ASSOC., INC.
C           THIS ROUTINE CALLS A SUBROUTINE 'SCACAL' FOR CALCULATION
C      OF THE ESP'S SCA VALUE, AND SUBROUTINE 'FIXATE' FOR COMPLETION OF
C      DRY SORBENT INJECTION'S COSTS WHERE APPPLICABLE.
C
C
C
      SUBROUTINE ESP
      INTEGER FMB,CT,BOTTOM,TYPCT,CLNLVL
      LOGICAL KO(2),TVA,LIMB
      REAL INDCST
      DIMENSION AC(13),CC(7),IC(6)
      COMMON /COSTS/ CAPCST(10,8),ANN(25),INDCST(10)
      COMMON /COAL1/ HHV,PS,PA,CST,PNA,ALK,CL,FEO,PM
      COMMON /STREAM/ PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
      COMMON /ESC/ IYC,RATPLT,RATLA2,RATMA2,RATOM
      COMMON /INPUT/BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,HHVCLN,PSCLN,PACLN,CSTCLN,PNACLN,ALKCLN,
     &CLCLN,FEOCLN,PMCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,WM,IAPCS(9),KO
     &,TVA
      COMMON /INSYSP/ CCEFF,UTEFF,BNHR,FANLOS,QIN,ALKIN
      COMMON /PARM/ P(500)
      COMMON /MATBAL/ TABLE(8,8)
      COMMON /OPT/ BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
      COMMON /FAN/ DELTAP,CSTFAN
C
C
      IODB = 8
```

```
      IO6 = 6
      IO7 = 7
      DO 10 I=1,13
      AC(I) = 0.0
   10 CONTINUE
      DO 15 I=1,7
      CC(I) = 0.0
   15 CONTINUE
      DO 20 I=1,6
      IC(I)= 0.0
   20 CONTINUE
C
      WRITE(IODB,8000)
 8000 FORMAT(T2,'ENTERING SUBROUTINE ESP')
      WRITE(IODB,8010) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,
     &                 PPHNOX,EFFESP,BSIZE,CF,(CAPCST(5,I),I=1,13),
     &                 INDCST,ANN,(P(J),J=91,110)
 8010 FORMAT (T10,'PPH,ACFM,TEMP,PPHMST,PPHALK=',5F14.4,
     &        /T10,'PPHPRT,PPHSO2,PPHNOX,EFFESP,BSIZE,CF=',6F14.4,
     &        /T10,'CAPCST(EST)=',8F10.2,
     &        /T10,'INDCST=',8F10.2,
     &       /T10,'ANN=',7F10.2,
     &        /T14,6F10.2,
     &        /T14,6F10.2,
     &        /t14,6f10.2,
     &        /T10,'P(91-110)=',10F10.2,
     &        /T19,10F10.2)
C
C*******************************************************************************
C
C      PART 1:  SIZING CALCULATIONS.
C
C      THIS PART CALCULATES SIZING PARAMETERS FOR USE LATER IN THE
C      PROGRAM.
C
C      SET ESP REMOVAL EFFICINCY TO THE MAXIMUM (PARAMETER 91) IF NO
C      VALUE HAS YET BEEN SET.
C
      IF (EFFESP .EQ. 0.) EFFESP = P(91)
C
C      IF NO NA2O CONTENT OF THE COAL WAS SET ON INPUT, USE THE DEFAULT
C      VALUE (PARAMETER 92).
C
      IF(PNA .EQ. 0.) PNA=.4
C
C     SET DISPOSAL FACTOR; ACCOUNT FOR DSI
C
      DISFCT = 1.0
      DO 500 I=1,9
      IF (IAPCS(I) .EQ. 7) DISFCT = .334
  500 CONTINUE
C
```

```
C       DETERMINE SCA (SPECIFIC COLLECTION AREA) IN SQUARE FEET OF PLATE
C       AREA PER 1000 ACFM. THIS IS DONE USING THE SUBROUTINE 'SCACAL'.
C
        CALL SCACAL (SCA,RESIS)
C
C       CALCULATE THE ASH RATE IN TONS PER HOUR.
C       ** NO BOILER ASH FOR UPGRADE **
        LIMB= .FALSE.
        DO 111 I=1,9
        IF(IAPCS(I) .EQ. 2) THEN
          LIMB= .TRUE.
        ENDIF
  111   CONTINUE
        IF(.NOT. LIMB .OR. NR .NE. 2 .OR. P(130) .EQ. 0.) THEN
          ASHRT = EFFESP/100. * PPHPRT / 2000.
        ELSE
          ASHRT = EFFESP/100. * (PPHPRT-TABLE(1,6))/2000.
        ENDIF
C
C       DEBUG OUTPUT.
C
        WRITE(IODB,8100) EFFESP,PNA,SCA,ASHRT
 8100   FORMAT(T10,'EFFESP,PNA,SCA,ASHRT=',4F14.4)
C
C*******************************************************************************
C
C       PART 2:  MODULE SPECIFIC OUTPUT.
C
C       THIS PART WITES THE MODUL,E SPECIFIC OUTPUT FOR THE FABRIC
C       FILTER.
C
C       WRITE HEADER.
C
        IF(KO(1)) WRITE(IO6,6200)
        IF(KO(2)) WRITE(IO7,6200)
 6200   FORMAT(///T5,'ELECTROSTATIC PRECIPITATOR (ESP)')
C
C       WRITE DETAIL MODULE-SPECIFIC OUTPUT.
C
        IF(KO(1)) WRITE(IO6,6210) EFFESP,SCA
        IF(KO(2)) WRITE(IO7,6210) EFFESP,SCA
 6210   FORMAT(T10,'THE ESP IS DESIGNED TO REMOVE ',F4.1,'% OF THE ',
     &             'PARTICULATE LOADING',
     &        /T10,'WITH A CALCULATED SCA OF ',F6.0,' THE ESP REFLECTS',
     &             ' A RIGID FRAME',
     &        /T10,'DESIGN AND A SUPERFICIAL GAS VELOCITY OF 4 ',
     &             'FT/S.')
        IF(KO(1)) WRITE(IO6,6215) RESIS
        IF(KO(2)) WRITE(IO7,6215) RESIS
 6215   FORMAT(T10,'A RESISTIVITY OF ',1PE7.1,' OHM-CM WAS USED')
C
C*******************************************************************************
```

```
C
C       PART 3:  DIRECT CAPITAL COST CALCULATION.
C
C       THIS PART CALCULATES THE DIRECT CAPITAL COSTS FOR THE ESP.
C       THESE INCLUDE THE ESP ITSELF, THE DUCTING, AND ASH HANDLING
C       SYSTEM. AN INSTALLATION FACTOR (PARAMETER 93) IS INCLUDED
C       WHERE NEEDED.
C
C>>>    IF A LIMB RETROFIT, USE ONLY ESP UPGRADE COSTS (SET IN LIMB)
C>>>    IF THE USER SO INDICATES VIA P(130).
C
      IF(.NOT. LIMB .OR. NR .NE. 2 .OR. P(130) .EQ. 0.) THEN
C
C         CALCULATE THE COST OF THE ESP (CC(1) . IT IS BASED ON ACFM
C         AND SCA.
          CC(1) = (8.17 * (SCA * ACFM/1000.) +232349.) * P(93)
C
C         CALCULATE THE COST OF DUCTING (CC(2)). NOTE, DUCTING FOR A
C         BOILER SIZED IN THE RANGE OF 250 TO 499 MW HAS A COST
C         BASED ON ACFM AND SCA. BOILERS INT H RANGE OF 500 TO 1300
C         MW USE TIS ALGORITHM MULTIPLIED BY AN ADDITIONAL DUCT COST
C         FACTOR (PARAMETER 94).
C
          CC(2) = (54.3*ACFM**.5 - 186.9 + (444.9 + 0.172*ACFM/SCA)*
     &          (.543*ACFM**.5 - 2.67)) *P(93)
          IF(BSIZE .GE. 500.) CC(2) = CC(2) * P(94)
C
C         CALCULATE THE COST OF ASH HOPPERS (CC(3)).
C
          CC(3) = 197.6 * BSIZE
C
C         CALCULATE THE COST OF PNEUMATIC PIPELINE (CC(4)).
C
          CC(4) = 22480. * ASHRT**.36
C
C         CALCULATE COST OF VACUUM (CC(5)).
C
          CC(5) = 50310. * ASHRT**.279
C
C         CALCULATE COST OF SILO BAGHOUSE (CC(6)) AND SILOS (CC(7))>
C         PARAMETER 95 IS A SIZING FACTOR FOR THE SIL.,OS AND N IS THE
C         NUMBER OF SILOS REQUIRED.
C
          N = IFIX(ASHRT/P(95)) + 1
          CC(6) = N * 29620. * ASHRT**.662
          CC(7) = N * 503100. * ASHRT**.279
C
C         CALCULATE SUBTOTALS FOR 'ESP & DUCTING', 'ASH HANDLING' AND
C         'TOTAL TOTAL ESP COST'.
C
          CAPCST(5,2) = (CC(1) +CC(2))*RATPLT*P(294)
          CAPCST(5,3) = (CC(3) + CC(4) + CC(5) + CC(6) + CC(7))*RATPLT*
```

```
     &  P(294)
        CAPCST(5,1) = CAPCST(5,2) + CAPCST(5,3)*P(294)
C
       ELSE
         IF(KO(1)) WRITE(IO6,10002)
         IF(KO(2)) WRITE(IO7,10002)
10002    FORMAT(T10,'ONLY THE COST OF THE UPGRADE DUE TO LIMB IS ',
     &  'INCLUDED IN THIS SCENARIO')
C
       ENDIF
C
C       DEBUG OUTPUT
C
       WRITE(IODB,8300) CC,(CAPCST(5,I),I=1,3)
 8300 FORMAT(T10,'CC=',7F12.2,
     &       /T10,'CAPCST(ESP)=',3F12.2)
C
C*******************************************************************************
CC
C       PART 4: INDIRECT CAPITAL COST CALCULATIONS.
C
C       THE INDIRECT CAPITAL COSTS ARE GENERALLY A PERCENTAGE OF THE
C       DIRECT CAPITAL COST OF DIRECT PLUS INDIRECT CAPITAL COSTS. THE
C       PERCENTAGES ARE RETAINED IN THE PARAMETER FILE (PARAETERS 96
C       THRU 101). EACH INDIRECT COST FOR EACH MODULE IS SUMMED INTO
C       THE ARRAY 'INDCST'.
C
C       CALCULATE ENGINEERING DESIGN AND SUPERVISION(TVA) OR
C       GENERAL FACILITIES(EPRI)
C
      IC(1) = P(97)/100. * CAPCST(5,1)
      INDCST(1) = INDCST(1) + IC(1)
C
C       CALCULATE A-E CONTRACTOR(TVA) OR ENG. AND HOME OFF. FEES (EPRI)
C
      IC(2) = P(96)/100. * CAPCST(5,1)
      INDCST(2) = INDCST(2) + IC(2)
C
C       CALCULATE CONSTRUCTION EXPENSE (TVA) PROJECT CONTINGENCY (EPRI)
C
      IC(3) = P(98)/100. * CAPCST(5,1)
      INDCST(3) = INDCST(3) + IC(3)
C
C       CALCULATE CONTRACTOR FEES (TVA) OR PROCESS CONTINGENCY (EPRI)
C
      IC(4) = P(99)/100. * CAPCST(5,1)
      INDCST(4) = INDCST(4) + IC(4)
C
C       SUM DIRECT AND INDIRECT CAPITAL COSTS TO HAVE A BASE FOR THE
C       REMAINDER OF THE INDIRECT COSTS.
C
     COST = CAPCST(5,1) + IC(1) + IC(2) + IC(3) + IC(4)
```

```
C
C       CALCULATE CONTINGENCY (TVA) OR SALES TAX (EPRI)
C
        IF (TVA) THEN
              IC(5) = P(100)/100. * COST
           ELSE
              IC(5) = P(100)/100. * CAPCST(5,1)
           ENDIF
        INDCST(5) = INDCST(5) + IC(5)
C
C       INCLUDE IC(5) IN RUNNING COST BASE TOTAL
C
        COST = COST + IC(5)
C
C       CALCULATE ROYALTIES (TVA) OR ROYALTY ALLOWANCE (EPRI)
C
        IC(6) = P(101)/100. * COST
        INDCST(6) = INDCST(6) + IC(6)
C
C       CALCULATE INTEREST DURING CONSTRUCTION (TVA)
C
        IF (TVA) THEN
          IC(8) = P(138)/100. * COST
          INDCST(8) = INDCST(8) + IC(8)
        ENDIF
C
C         CALCULATE ALLOWANCE FOR STARTUP AND MODIFICATIONS (TVA) OR
C                   INITIAL CATALYST (EPRI)
C
        IC(9) = P(110)/100. * COST
        INDCST(9) = INDCST(9) + IC(9)
C
C         CALCULATE LAND (TVA AND EPRI)
C
        IC(10) = P(181)
        INDCST(10) = INDCST(10) + IC(10)
C
C       DEBUG OUTPUT.
C
       WRITE (IODB,*)IC,INDCST
 8400 FORMAT (T10,'IC=',6F12.2,
     &         /T10,'INDCST=',6F12.2)
C
C*******************************************************************************
C
C       PART 5:  ANNUAL OPERATING COST CALCULATIONS.
C
C       ANNUAL OPERATING COSTS ARE REPORTED FOR THE ENTIRE SYSTEM (ALL
C       MODULES). MOST PARAMETERS ARE KEPT BY QUANTITY. ALL VALUES
C       ARE ON A PER YEAR BASIS.
C
C       THE OPERATIONSL LABOR IS A FUNCTON OF GAS FLOW.
```

```
C       SUPERVISORY LABOR IS A PERCENTAGE (PARAMETER 102) OF THE OPERAT-
C       ING LABOR.
C
        IF(.NOT. LIMB .OR. NR .NE. 2 .OR. P(130) .EQ. 0.) THEN
        AC(1) = (3.51*ACFM**.4 + .00146*ACFM)*(1.+P(102)*.01)
        ANN(1) = ANN(1) + AC(1)
C
C       CALCULATE ELECTRICITY REQUIREMENT IN KILOWATTHOURS.
C
        HP= 113.99 * ALOG(ASHRT*2000.) - 1084.7
        IF(HP .LT. 50.) HP = 50.
        AC(4) = SCA * ACFM *.01752 * CF + (.746*8760.*CF*HP)
        ANN(4) = ANN(4) + AC(4)
C
C       CALCULATE WATER REQUIREMENT IN 1000 GALLONS . NOTE, A DISPOSAL
C       FRACTION 'DISFCT' IS DETERIMED BASED ON THE USE OF A LIME
C       SPRAY DRYER. PARAMETER 103 IS THE PROPORTION OF WATER ADDED TO
C       THE ASH.
C
        AC(5) = 2100.72 * DISFCT * ASHRT * CF * (P(103)/100.)
        ANN(5) = ANN(5) + AC(5)
        ENDIF
C
C       NO REHEAT IS REQUIRED.
C
C       NO CHEMICALS ARE REQUIRED.
C
C       CALCULATE MAINTENANCE MATERIAL AND LABOR COST.
C
        AC(10) = CAPCST(5,1) * P(109)*.01
        ANN(10) = ANN(10) + AC(10)
C
C       NO FUEL COST PREMIUMS.
C
C       ONLY DRY WASTES ARE PRODUCED. CALCULATE THIS DISPOSAL QUANTITY
C       IN TONS.
C
C
        AC(13) = DISFCT * (1.+P(103)/100.) * ASHRT * 8760. * CF
        ANN(13) = ANN(13) + AC(13)
C
C       DEBUG OUTPUT.
C
        WRITE(IODB,8500) AC,ANN
 8500 FORMAT(T10,'AC=',7F10.2,
     &       /T13,6F10.2,
     &       /T10,'ANN=',7F10.2,
     &       /T14,6F10.2,
     &       /T14,6F10.2,
     &       /T14,6F10.2)
C
C*************************************************************************
```

```
C
C       PART 6:  MATERIAL BALANCE CALCULATIONS.
C
C       AT THE END OF EACH MODULE, THE STREAM PARAMETERS CONTAINED IN
C       THE COMMON 'STREAM' ARE ADJUSTED FOR THE AFFECTS OF THIS MODULE.
C
C       CALCULATE NEW FLUE GAS ALKALINITY.
        PPHALK = PPHALK - EFFESP*.01*PPHALK
C
C       CALCULATE NEW FLUE GAS PARTICULATES.
C
        OPHPRT= PPHPRT
        PPHPRT = PPHPRT - EFFESP/100. * PPHPRT
C
C       CALCULATE NEW FLUE GAS SO2. FIRST, DETERMINE SO2 EFFICIENCY
C       BASED ON PRECEEDING MODULES.
C
        SO2EFF = 0.0
        DO 600 I=1,9
        IF(IAPCS(I) .EQ. 2) SO2EFT = P(105)/100.
        IF(IAPCS(I) .EQ. 4) SO2EFT = P(106)/100.
        IF(IAPCS(I) .EQ. 7) SO2EFT = P(107)/100.
        IF(IAPCS(I) .EQ. 9) SO2EFT = P(108)/100.
        IF(SO2EFT .GT. SO2EFF) SO2EFF = SO2EFT
  600 CONTINUE
        OPHSO2 = PPHSO2
        PPHSO2 = OPHSO2 * (1.-SO2EFF)
C
C       CALCULATE NEW FLUE GAS FLOW (LBS/HR)
C
        PPH = PPH - EFFESP*.01*PPHPRT - OPHSO2*SO2EFF
C
C       NO CHANGE TO FLUE GAS ACFM, TEMPERATURE, MOISTURE, OR NOX.
C
C       ADJUST OVERALL PRESSURE DROP DUE TO THIS MODULE
C
        IF(.NOT. LIMB .OR. NR .NE. 2 .OR. P(130) .EQ. 0.) THEN
           DELTAP = DELTAP +P(104)
        ENDIF
C
C       DEBUG OUTPUT
C
        WRITE(IODB,8600) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,
     &                   PPHNOX
 8600 FORMAT (T10,'PPH,ACFM,TEMP,PPHMST=',4F14.4,
     &        /T10,'PPHALK,PPHPRT,PPHSO2,PPHNOX=',4F14.4)
C
C*******************************************************************************
C
C*******************************************************************************
C
C       PART 7: FIXATION CONSIDERATIONS.
```

```
C
C     IF A DRY SORBENT INJECTION MODULE PRECEEDS THE ESP, THE WASTES
C     PRODUCED MUST BE FIXATED DUE TO NAHCOLITE CONTAMINATION. THIS
C     PART ADJUXTS THE PREVIOUSTLY CALCULATED WATES AND THEN CALLS
C     THE SUBROUTINE 'FIXATE' TO COMPLETE THE DSI COSTS.
C
C     DETERMINE IF DSI HAS PRECEEDED THIS MODULE. IF NOT, SKIP TO END.

      DO 700 I=1,9
      IF(IAPCS(I) .EQ. 9) GO TO 710
  700 CONTINUE
      GO TO 800
C
C     REMOVE THE PREVIOUSLY CALCULATED DRY WASTES FROM THE ANNUAL
C     COSTING ARRAY.
C
  710 ANN(13)= ANN(13)-AC(13)
C
C     CALCULATE THE SO2 AND ASH IN TONS/HOUR REMOVED BY THIS ESP.
C
      SO2= (OPHSO2-PPHSO2)/2000.
      ASH= (OPHPRT-PPHPRT)/2000.
C
C     CALL SUBROUTINE 'FIXATE' TO FINISH DSI COSTS.
C
      CALL FIXATE(SO2,ASH)
C
C      PART 8:   CONCLUSION.
C
  800 WRITE(IODB,8800)
 8800 FORMAT(T2,'LEAVING SUBROUTINE ESP.')
C
      RETURN
      END
```

```
      BLOCK DATA
C
      CHARACTER*4 IY
      REAL OD(6),XL(6),XT(6),FACTOR(6),NFT(6),CPS(6),CPH(6),A1,A2
     &,A3,A4,EPS,M(6),MDUST(10),MWETGS(14),MDUST1(10)
      REAL CIPU(11,4),TTOL,XMW(10),A(32),B(32),C(32),D(32)
      REAL MH2O,MCA,MCAO,MMG,MMGO,MCL,MCO3,MCACO3,MSO2,MSO3,MSO4,
     &MCASO3,MCASO4,XSCRUB(24),NTANKS
      DIMENSION PIPIN1(80),PIPIN2(80)
      INTEGER ITMAX,MWGT(4)
      DIMENSION ELECT(10,7,2),RHP(10),GPMSP(9),PERPMP(9)
      DIMENSION BLKA(11),BLKB(11),BLKC(41),BLKCC(11),BLKU(11)
C
      COMMON/HOPE/MDUST,OD,XL,XT,FACTOR,NFT,CPS,CPH,A1,A2,A3,A4,
     &EPS,M
      COMMON/COUPLE/IY,CIPU
      COMMON/NTHREE/NUMRAW,NUMUTL,NUMUNT
      COMMON/GASHOT/ITMAX,TTOL,MDUST1
      COMMON/BALMAT/MH2O,MCA,MCAO,MMG,MMGO,MCL,MCO3,MCACO3,MSO2,
     &MSO3,MSO4,MCASO3,MCASO4
      COMMON/MODSR/C1,C2,C3,C4,C5,C6,D1,D2,D3,V1,V2,V3,W1,W2,W3,W4
     &,NTANKS,DS
      COMMON/CALLEQ/XMW
      COMMON/QEB/A,B,C,D
      COMMON/CALCK/BLKA,BLKB,BLKC,BLKCC,BLKU
      COMMON/STRAEM/MWETGS
      COMMON/TEQUIP/XSCRUB
      COMMON/PUMPSL/ELECT,RHP,PERPMP,GPMSP
      COMMON/PIPER/PIPIN1,PIPIN2
      COMMON/ISI/MWGT
C
      DATA MDUST/44.01,36.46,64.06,32.,28.02,18.016,12.01,35.45,
     1  32.06,1.008/
      DATA MDUST1/44.01,36.46,64.06,32.,28.02,18.016,12.01,35.45,
     1  32.06,1.008/
      DATA OD/1.,1.,2.375,3.5,4.5,6.625/
      DATA XL/2.,2.,1.6842,1.714,1.777,1.811/
      DATA XT/2.,2.,1.6842,1.714,1.777,1.811/
      DATA FACTOR /2.,2.,2.46,2.40,2.29,2.23/
      DATA NFT/6.,6.,3.,2.,1.5,1./
      DATA CPS /0.2141,0.1913,0.1487,0.2066,0.2437,0.4670/
      DATA CPH /0.2365,0.1935,0.1575,0.2291,0.2471,0.4731/
      DATA A1,A2,A3,A4/2.8E-4,2.1E-5,2.46,.62/
      DATA EPS/.001/
      DATA  M/44.01,36.46,64.06,32.,28.02,18.016/
C
      DATA IY/'Y'/
      DATA CIPU/261.2,312.0,222.7,90.3,297.0,323.9,242.4,95.0,
     &          314.0,336.2,263.9,100.0,316.9,337.6,265.1,105.0,
     &          324.4,345.9,265.5,110.3,340.6,363.2,278.8,115.8,
     &          357.7,381.4,292.7,121.6,375.5,400.4,307.3,127.6,
     &          394.3,420.4,322.7,134.0,414.0,441.5,338.9,140.7,
```

```
     &          434.7, 463.5, 355.8, 147.7/
C
      DATA NUMRAW, NUMUTL, NUMUNT/12, 8, 23/
C
      DATA ITMAX, TTOL/20, 0.5/
C
      DATA MH2O, MCA, MCAO, MMG, MMGO, MCL/18.016, 40.08, 56.08, 24.31, 40.31,
     1     35.45/
      DATA MCO3, MCACO3/60.01, 100.09/
      DATA MSO2, MSO3, MSO4, MCASO3, MCASO4/64.06, 80.06, 96.06, 129.15, 172.15/
C
      DATA C1, C2, C3, C4, C5, C6/1.2, 1.5, 0.75, -8.8E-5, -0.12, 1./
      DATA D1, D2, D3/6.1E-5, 0.65, 1.5/
      DATA DS/1.5/, NTANKS/1./
      DATA V1, V2, V3/3.66E-3, 0.60, 1.20/
      DATA W1, W2, W3, W4/1.5E-3, 1., 0.40, 0.19/
C
      DATA XMW/40.08, 24.31, 22.99, 39.10, 80.06, 96.06, 60.01, 35.45, 146., 0.0/
C
      DATA  A/4470.99, 0., 3404.71, 2902.39, -843.67, -633.84, 475.14,
     1 -273., -301.85, -475.48, -504.8, 2572.1, -517.99, -235.08, -504.8,
     2 -432.5, -1057.86, -1245., 0., 0., 0., -303.41, -241., -1660., 0., 4944.0,
     3 530.49, -955.14, -480.85, 0., 0., 0./
C
      DATA  B/0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 23.15, 0., 0., 0., 0., 0.,
     1 0., 0., 0., 0., 0., 0., 0., 0., 37.745, -12.9722, 0., 0., 0., 0., 0./
C
      DATA  C/.01706, 0., .032786, .02379, 0., 0., .018222, 0., 0., 0., 0.,
     1 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., .032331, 0., 0.
     2 , 0., 0., 0./
C
      DATA  D/6.0875, 0., 14.8435, 6.4980, -4.7171, -9.3320, 5.0435, -2.29,
     1 -2.272, -4.7954, -5.091, 63.60, -4.3223, -1.747, -5.091, -4.3715, -5.7950
     2 , -4.49, .40, .57, .25, -2.2852, -1.529, -13.88, -6.347, 105.36, -25.7085,
     3 -7.871, -12.514, -4.420, -5.495, -2.602/
C
      DATA  BLKA/6., 3., 3., 4.5, 4.5, 3., 4.5, 3., 2., 5., 4./
      DATA  BLKB/.4, .3, .3, 0., 0., 0., .1, .3, -.2, .1, 0./
      DATA  BLKC/1., 1., 0., 1., 4., 0., 1., 4., 1., 4., 4., 1., 1., 0., 0., 0.,
     1 4., 1., 1., 0., 0., 0., 1., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0.
     2 , 1., 2., 0./
      DATA  BLKCC/1., 1., 0., 1., 4., 4., 4., 4., 1., 1., 1./
      DATA  BLKU/0., 0., .076, 0., 0., 0., 0., 0., 0., 0., 0./
C
      DATA  MWETGS/129.1, 172.1, 100.1, 0., 18.016, 40.08, 24.31, 80.06, 96.06,
     1 35.45, 146., 40.31, 56.08, 74.1/
C
      DATA XSCRUB/7089.7, 4006.9, 96.92, 214.958, 161.219, 36.01,
     &           62.239, 2.4126, .019413, 484.306, 3.6628, 4.4708,
     &           .64497, .70419, 1.03545, .9972, .99723, 1.09583,
     &           .6765, .3008, 1.0438, 1.04483, .81783, .64795/
C
```

```
      DATA ELECT/355.,370.,524.,2*762.,2*1672.,2*3785.,6375.,
     &           0.20,0.27,0.42,6*0.89,0.00205602,
     &           0.92,0.96,1.02,6*1.10,0.00514005,
     &           0.72,0.86,1.16,6*1.86,0.0076956,
     &           7*1.,2.,2*3.,
     &           190.,300.,510.,900.,2*1790.,3*70.7362,471.027,
     &           5*186.311,3*0.27048,2*204.728,
     &           10*0.0,
     &           9*0.0,1.00048,
     &           9*0.0,1.00048,
     &           9*0.0,0.948743,
     &           10*0.0,
     &           6*0.0,3*0.89287,0.639446,
     &           5*0.33909,3*1.80974,2*0.855266/
      DATA RHP/2.5,7.5,15.,35.,50.,90.,133.,233.,250.,1500./
      DATA GPMSP/115.,290.,450.,870.,1160.,2650.,4250.,7000.,20001./
      DATA PERPMP/1775.,2820.,4018.,4457.,7410.,8178.,13483.,21412.,
     &29968./
C
      DATA PIPIN1/7489.36,320.104,4552.33,689.75,3261.88,3380.67,
     &1826.82,87.647,2357.58,613.53,26670.4,17591.7,3364.15,971.743,
     &21926.26,14483.4,3364.15,971.74,1183.11,589.40,4145.32,1357.90,
     &436.95,969.2,1333.54,828.2,283.8,63.513,2389.58,11615.0,
     &4252.19,11754.5,700.38,666.88,1.3659,2.807,3.88,1.664,
     &16240.,3734.5,3741.6,584.32,1486.05,1272.79,3741.6,584.32,
     &1486.05,1272.79,3.88,1.66,150.34,54.62,1075.64,360.75,
     &713.25,223.564,3.88,1.66,3865.73,1476.03,3.88,1.66,
     &44.36,224.83,1771.26,1139.47,1.3655,2.8073,2084.93,1281.68,
     &2522.47,3674.84,43.88,22.445,150.34,54.621,1075.64,
     &360.75,713.247,215.94/
      DATA PIPIN2/.257947,.4528,.469269,.280447,.416544,.399396,
     &.353308,.396298,.177282,.19267,.43751,.262947,.272633,.188209,
     &.361478,.286086,.272633,.188209,.443309,.245169,.389936,.303329,
     &.495443,.300235,.405396,.278248,.730249,.649843,.58499,.585426,
     &.33493,.3292,.414571,.255696,.41857,.2708,.378423,.308282,
     &.235404,.148409,.478149,.40712,.583853,.376378,.478149,.40712,
     &.583853,.376378,.378423,.308282,.541782,.406928,.527415,.411663,
     &.48411,.348086,.378423,.308282,.355653,.253727,.378423,.308282,
     &.4985,.2694,.412268,.302154,.41857,.2708,.486484,.270809,
     &.388316,.340924,.481095,.339238,.541782,.406928,.527415,
     &.41163,.48411,.348086/
C
      DATA MWGT/100,184,74,132/
C
      END
```

```
C
      SUBROUTINE PDUMP(LAB1)
      CHARACTER*64 LAB,BLANK,QUEST,LAB1(500)
      CHARACTER*4 FILEN(2),EXT(2)
      CHARACTER*8 FILE2
      CHARACTER*12 FILE1
      INTEGER Y,Z,Z1,FORM,IAPCS(9),FMB,CT,BOTTOM,TYPCT,CLNLVL,YY(500)
      LOGICAL EX,TVA,KO(2),CHANGE(500),ANY
      COMMON /PEDIT/ CHANGE,FILE1
      COMMON /INPUT/    BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,
     &          PNAROM,ALKROM,CLROM,FEOROM,PMROM,VROM,FCROM,HHVCLN,PSCLN,
     &                 PACLM,CSTCLN,PNACLN,ALKCLN,CLCLN,FEOCLN,PMCLN,
     &          VCLN,FCCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,IAPCS,KO,
     &                 TVA
      COMMON /PARM/ P(500)
C
C     REPORT PARAMETER FILE CHANGES
C
C>>>  LOOP ON OUTPUT DEVICE
      DO 850 LD= 6,7
         IF( .NOT. KO(LD-5)) GOTO 850
         WRITE(LD,20000) FILE1
20000    FORMAT(1H1,//T14,
     &   'INTEGRATED AIR POLLUTION CONTROL SYSTEM COSTING PROGRAM',/T13,
     &    57(1H-),//T27,'USER INPUT SUMMARY (CONTINUED)',/T26,32(1H-)
     &    //T25,'PARAMETER FILE USED: ',A)
         ANY= .FALSE.
         DO 825 I=1,500
            IF(CHANGE(I)) THEN
               IF(.NOT. ANY) THEN
                  WRITE(LD,21000)
                  ANY= .TRUE.
               ENDIF
               WRITE(LD,22000) P(I),LAB1(I)
            ENDIF
 825     CONTINUE
         IF(.NOT. ANY) WRITE(LD,23000)
21000    FORMAT(//T10,'THE FOLLOWING CHANGES WERE MADE TO THIS PARAMETER
     & FILE FOR THIS RUN:'//'  NEW VALUE   DESCRIPTION'/' ---------   ',
     &   64(1H-))
22000    FORMAT(1H ,G10.4,3X,A)
23000    FORMAT(//T10,'NO CHANGES WERE MADE TO THIS PARAMETER FILE FOR',
     &          ' THIS RUN.')
 850  CONTINUE
      IF(KO(2)) WRITE(7,'(1H1)')
      RETURN
      END
```

```
$DEBUG
$large
      SUBROUTINE BATCH(FNAME,EXB)
C
CC
CC            THIS SUBROUTINE ALLOWS USER TO CALL A BATCH FILE FOR
CC       USE AS INPUT.  THE BATCH FILE SHOULD BE PREPARED AND EDITED
CC       USING LOTUS 1-2-3 OR A SUITABLE WORD PROCESSOR. THE FILENAME IS
CC       REQUESTED IN THE INPUT SUBROUTINE.
CC
CCC                  WRITTEN BY: PEI ASSOCIATES, INC. (JW)
CC                                (12-17-84)
C
       CHARACTER*4 AFMB,ACLNLV,ANR,ABOT,AIAPCS,AY,AN,DUMMY,OPTMZ
       INTEGER FMB,CT,BOTTOM,TYPCT,CLNLVL
       LOGICAL VALID,KO(2),TVA,EXB
       DIMENSION RINPEQ(26),INPEQ(17),TYPCC(6,22),LAB(300,17)
       DIMENSION AFMB(3,2),ACLNLV(8,2),ANR(2,2),ABOT(2),AIAPCS(8,9)
       COMMON /PARM/ P(500)
       COMMON /COMM/ COMM1,COMM2
       COMMON/INPUT/BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
      &ALKROM,CLROM,FEOROM,PMROM,VROM,FCROM,HHVCLN,PSCLN,PACLN,CSTCLN,
      &PNACLN,ALKCLN,CLCLN,FEOCLN,PMCLN,VCLN,FCCLN,FMB,NR,IS,CT,TYPCT,
      &CLNLVL,BOTTOM,NM,IAPCS(9),KO,TVA
       COMMON /NEW/IFACT
       COMMON /BATCHC/OPTMZ,NOPT,TARGET,NEXTFL
       CHARACTER*13 FILE,FNAME,PARAM(8)*1,NEXTFL,COMM1*80,COMM2*80
       EQUIVALENCE (BSIZE,RINPEQ(1)),(FMB,INPEQ(1))
C
       DATA AN/'N'/
       DATA AY/'Y'/
       DATA TYPCC/ 11592.,11922.,12058.,10359., 8789., 7500.,
      &              2.23,  3.43,  0.89,  4.27,  0.56,  0.94,
      &             15.9,  13.0, 16.60, 20.60,  8.15,  5.90,
      &              0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
      &              0.4,   0.4,   0.4,   0.4,   2.8,   4.4,
      &              6.5,   6.5,   6.5,   6.5,  20.9,  31.9,
      &              0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
      &              9.0,   9.0,   9.0,   9.0,   9.0,   9.0,
      &              3.3,   5.0,   3.5,   9.6,  25.2,  32.0,
      &33.8,37.2,35.4,40.6,36.4,42.0,47.0,44.8,43.0,29.2,30.3,20.1,
      &            12596.,12845.,13611.,11507., 9050., 7840.,
      &             1.42,  2.74,  0.83,  3.50,  0.43,  0.54,
      &             10.0,  66.6,   4.6,  10.7,  6.46,   5.3,
      &             4.57,  3.85,  4.02,  3.74,  3.87,  3.82,
      &              0.4,   0.4,   0.4,   0.4,   2.8,   4.4,
      &              6.5,   6.5,   6.5,   6.5,  20.9,  31.9,
      &              0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
      &              9.0,   9.0,   9.0,   9.0,   9.0,   9.0,
      &              5.6,   4.4,   5.4,  11.0,  24.0, 30.01,
      &33.9,40.2,36.7,43.6,36.1,42.0,50.5,48.8,53.3,34.7,33.4,22.7/
C
C      CHECK FOR EXISTENCE OF BATCH FILE
```

```
C
      INQUIRE(FILE=FNAME,EXIST=EXB)
      IF( .NOT. EXB) THEN
        WRITE(6,1000) FNAME
 1000   FORMAT(' Batch file does not exist: ',A)
        RETURN
      ENDIF
C
C     READ BATCH FILE
C
      OPEN (15,FILE=FNAME)
   30 FORMAT (A1)
      READ (15,30) DUMMY
      READ (15,30) DUMMY
C
C     READ IN COMMENT RECORDS
C
      READ(15,'(A)') COMM1
      READ(15,'(A)') COMM2
C
C     READ IN TVA/EPRI FORMAT
C
      READ (15,*) FORM
      TVA=.TRUE.
      IF (FORM.EQ.2) TVA=.FALSE.
C
C     READ IN PARAMETER FILENAME
C
      READ (15,40) PARAM
   40 FORMAT (8A1)
      CALL PARMB(PARAM)
C
C    READ IN FIRING CONFIGURATION (WALL/TANGENTIAL)
C
      READ(15,*)   FMB
C
C    READ IN BOILER SIZE (IN MWE)
C
      READ(15,*)   BSIZE
C
C    READ IN BOILER CAPACITY FACTOR (%)
C
      READ(15,*) CF
      CF=CF*.01
C
C    READ IN CONSTRUCTION STATUS (NEW/RETROFIT)
C
      READ(15,*) NR
C
C    READ IN DATE OF OPERATION
C
      READ(15,*) IS
```

```
C
C     READ IN TEMPERATURE AT AIR HEATER EXIT
C
      READ(15,*) AHXT
C
C     READ IN ACFM AT AIR HEATER EXIT
C
      AHFX=P(58)
C
C     READ IN COAL CATEGORY (TYPICAL COAL/SPECIFIC CHARACTERISTS)
C
      READ(15,*) I
      IF (I.EQ.1) GO TO 100
C
C     USER INPUT COAL CHARACTERISTICS
C
C>>>> ROM COAL
  190 CLNLVL=1
C
C     READ IN COAL TYPE (BIT/SUB/LIG)
C
      READ(15,*)  CT
      IFACT=14
      IF(CT.EQ.3) IFACT=18
C
C     READ IN ROM HEATING VALUE
C
      READ(15,*)  HHVROM
C
C     READ IN SULFUR CONTENT OF ROM COAL
C
      READ(15,*) PSROM
C
C     READ IN ASH CONTENT OF ROM COAL
C
      READ(15,*) PAROM
C
C     READ IN ROM COAL COST ($/TON)
C
      READ(15,*) CSTROM
C
C     READ IN NA2O CONTENT OF ROM COAL
C
      READ(15,*) PNAROM
C
C     READ IN ROM COAL ALKALINITY
C
      READ(15,*)  ALKROM
C
C     READ IN CHLORINE CONTENT OF ROM COAL
C
      READ(15,*) CLROM
```

```
C
C     READ IN ROM COAL FE203 CONTENT
C
      READ(15,*) FEOROM
C
C     READ IN MOISTURE CONTENT OF ROM COAL
C
      READ(15,*)  PMROM
C
C     READ IN VOLATILE MATTER CONTENT
C
      READ (15,*) VROM
      READ (15,*) FCROM
C
C     READ IN COAL CLEANING (Y/N)
C
      READ(15,30) A
C>>>> CLEAN COAL
      IF(A .EQ. AN) GO TO 200
      CLNLVL= 2
C
C     READ IN HEATING VALUE OF CLEAN COAL
C
      READ(15,*) HHVCLN
C
C     READ IN CLEAN COAL SULFUR CONTENT
C
      READ(15,*) PSCLN
C
C     READ IN ASH CONTENT OF CLEAN COAL
C
      READ(15,*) PACLN
C
C     READ IN COST OF CLEAN COAL
C
      READ(15,*) CSTCLN
C
C     READ IN NA2O CONTENT OF CLEAN COAL
C
      READ(15,*) PNACLN
C
C     READ IN CLEAN COAL ALKALINITY
C
      READ(15,*) ALKCLN
C
C     READ IN CHLORINE CONTENT OF CLEAN COAL
C
      READ(15,*) CLCLN
C
C     READ IN FE203 CONTENT OF CLEAN COAL
C
      READ(15,*) FEOCLN
```

```
C
C     READ IN CLEAN COAL MOISTURE CONTENT
C
      READ(15,*) PMCLN
      GO TO 200
C
C     READ IN CLEAN COAL VOLATILE MATTER CONTENT
C
      READ (15,*) VCLN
      READ (15,*) FCCLN
C
C     READ IN WHICH TYPICAL COAL TYPE (1-6)
C
  100 READ(15,*) TYPCT
      IFACT=14
      IF (TYPCT.EQ.6) IFACT=18
C
C     SET TYPICAL COAL CHARACTERISTICS
C
      DO 625 I=5,26
  625 RINPEQ(I)= TYPCC(TYPCT,I-4)
C
C     READ IN COAL CLEANING LEVEL (ROM/PHYSICAL CLEANING)
C
      READ(15,*) CLNLVL
C
C     READ IN BOTTOM ASH CONFIGURATION (DRY/WET)
C
  200 READ(15,*) BOTTOM
C
C     READ IN OUTPUT OPTION (1-3)
C
      READ(15,*) IO
      IF(IO.NE. 2) KO(2)=.TRUE.
      IF(IO .GE. 2) KO(1)=.TRUE.
C
C     READ IN NUMBER OF OPTIONS (1-9)
C
      READ(15,*) NM
      DO 300 I=1,NM
C
C     READ IN OPTION NUMBER
C
      READ(15,*) IAPCS(I)
  300 CONTINUE
C
C     READ IN OPTIMIZE OPTION (Y/N)
C
      READ (15,30) OPTMZ
      WRITE (6,*) OPTMZ
      IF(OPTMZ.NE.AY) GO TO 420
C
```

```
C     READ IN MODULE TO BE OPTIMIZED
C
      READ (15,*) NOPT
      DO 400 I=1,NM
         IF(NOPT.EQ.IAPCS(I)) GOTO 410
  400 CONTINUE
      OPTMZ=AN
  410 CONTINUE
C
C     READ IN TARGET EMISSION RATE
C
      READ (15,*) TARGET
C
C     READ IN NEXT BATCH FILENAME
C
  420 READ (15,50) NEXTFL
   50 FORMAT (A13)
      RETURN
      END
```

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     FUNCTION SUBPROGRAM TO CONVERT NUMERIC INTEGER INTO REAL DISPLAY
C     REVISION DATE 77-04-26
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      REAL FUNCTION RELDSP(INT)
      CHARACTER*4 STARS
      DATA STARS/'****'/
      RELDSP=STARS
      IF (INT.GT.999) RETURN
      I=INT
      J=I/10
      I=I-10*J
      K=J/10
      IF (K.GT.9) RETURN
      J=J-10*K
      INTDSP=I+240+(J+240)*256+(K+240)*256*256
      RELDSP=INTDSP
      RELDSP=RELDSP/256./256./256.
      RETURN
      END
```

```
$DEBUG
$NOFLOATCALLS
$LARGE
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C      SUBPROGRAM BECHTL
C      REVISION DATE 83-01-21
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       SUBROUTINE BECHTL
C
       COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
      1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
      2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
      3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
      4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
      5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
      6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
       COMMON/BYPAS2/BYLBH(6), BYLBMH(6), BSCFM, BACFM, PASSBY, XPASSD, IBY
       COMMON/BOILER/MW, BHR, EXSAIR, RMW, PREC, SPASH, WPRITE, KCLEAN, SMRW, SMCL
       COMMON/COAL/HVC, WPC, WPH, WPO, WPN, WPSUL, WPCL, WPASH, WPH2O, SULO, ASHO
       COMMON/EMIT/ISO2, XSO2, ISSO2, ELISO2, SO2REF, SO2LMB, SO2PPM
       REAL MFLOW(20,12), WFLOW(20,12), TOTFLO(20), GPM(20)
       COMMON/FLOWS/GPM, MFLOW, WFLOW, TOTFLO
       COMMON/LS/IALK, ISR, SR, SRADD, WPMGO, WPI, WPM, LSADD, LIMEAD, MGOADD, WMG
       COMMON/OUTPT/JOUTPT, JIT, JEQ1, JEQ2, JEQ3
       COMMON/SCRUB/ISCRUB, IFOX, IADD, VLG, VTR, IFAN
       COMMON/SCRUBR/LG, V, PPMOUT, SO2R, ELIMIT, PR, PD, WPS, OX, PH, NSTAGE,
      1              NGRID, HS, ACS, ACSTRN
       COMMON/SKGAS/SSCFM, SACFM, TSK, SDUST, SWDUST, SMPCT, SLBMH, SLBH, TGAS
       COMMON/SOLDIS/CASO3, CASO4, CACO3, INSOLS, WDIS, PSD, DENS, DEND
       COMMON/TRAN/NOTRAN, NOREDN, NTOTAL, PCNTRN, NORAW, NOUTIL, NOUNIT,
      1            NOPROD, NOSHIP, NSPREP, NSCOMP, NOCOMP, NOBYDU
       COMMON/WGAS/WACFM, WSCFM, TWG, ASHOUT, WASHOT, WMPCT, WLBMH, WLBH
       DIMENSION RMPCT(6), RHLBMH(6), RHLBH(6)
       REAL MW
       COMMON/COMPLE/ICOMPL, IRPT, IRUN, NEXT, IEQPR, IWTBAL, ITM, INPOPT, NOPART
       REAL LGMAX
       REAL LSADD, LIMEAD, MGOADD
       COMMON/RCOAL/HVCR, WPCR, WPHR, WPOR, WPNR, WPSULR, WPCLR, WPASHR, WPH2OR
       REAL LG
       REAL INSOLS
       REAL SMPCT(6), SLBMH(6), SLBH(6)
       REAL WMPCT(6), WLBMH(6), WLBH(6)
       REAL KGTA, KGEHT, KPCO2
       COMMON/HGAS/ACFM, THG, YCO2, YHCL, YSO2, YO2, YN2, YH2O, ASH, WCL, WASH, IESP
       COMMON/WGCOMP/YWCO2, YWHCL, YWSO2, YWO2, YWN2, YWH2O, WVAPOR, NTRAIN
       REAL NTRAIN
       REAL LMAX, L
       COMMON/H2OAMT/SEEPRT, EVAPRT, RAIN, SEEP, EVAP, EVAPC, RAFAL, RAFALC,
```

```
     1                WLSM,H2OHYD,TWA,TWR
      REAL MH2O,MCASO3,MCASO4,MUW
C
      REAL M(6),SCREM,PCTSCR,SOBOIL
      COMMON/SRHT/TSTEAM,HVS,VRH,WSTEAM,ARH,DPRH,URH,Q,NT,NB,PL,OD1,
     1             ARINCL,ARCORT,NINCLT,NCORTT,NINCLB,NCORTB,TEMPS
      DIMENSION ARH(6),DPRH(6),URH(6),NT(6),NB(6),OD1(6),ARINCL(6),
     1          ARCORT(6),NINCLT(6),NCORTT(6),NINCLB(6),NCORTB(6)
C
      DIMENSION OD(6),XL(6),XT(6),FACTOR(6)
      DIMENSION CPS(6),CPH(6)
      DOUBLE PRECISION DELT1,DELT2,TLN
      REAL KM,NFT(6),MUF,N1
      COMMON/FLYASH/IASH,ASHUPS,ASHSCR,ASHPRU,ASHPRS,ASHLBU,ASHLBS
      COMMON/HGAS2/MPCT,LBMH,LBH,SCFM,HHG,HWG,CFR,BCFR,RCFR
      REAL MPCT(6),LBMH(6),LBH(6)
      REAL MDUST(10),MFR(6),F(6),N2AIR,MWDG
      DATA MDUST/44.01,36.46,64.06,32.,28.02,18.016,12.01,35.45,
     1  32.06,1.008/
C
      DATA OD/1.,1.,2.375,3.5,4.5,6.625/
      DATA XL/2.,2.,1.6842,1.714,1.777,1.811/
      DATA XT/2.,2.,1.6842,1.714,1.777,1.811/
      DATA FACTOR /2.,2.,2.46,2.40,2.29,2.23/
      DATA NFT/6.,6.,3.,2.,1.5,1./
      DATA CPS /0.2141,0.1913,0.1487,0.2066,0.2437,0.4670/
      DATA CPH /0.2365,0.1935,0.1575,0.2291,0.2471,0.4731/
      DATA A1,A2,A3,A4/2.8E-4,2.1E-5,2.46,.62/
      DATA EPS/.001/
      DATA  M/44.01,36.46,64.06,32.,28.02,18.016/
C
C
      DO 11 I=1,6
      BYLBH(I)=0.0
      BYLBMH(I)=0.0
11    CONTINUE
      PHAA = PH
      ISUBSW=1
  990 FORMAT(' ****     BECHTL CALLING SUBROUTINE #',I3,' ****')
      DO 1000 I=1,20
      GPM(I)=0
      TOTFLO(I)=0
 1000 CONTINUE
      ISTAND = ISO2
      IT=0
      ITMAX=20
C     INITIALIZE L/G AT 50
      IF (IALK.EQ.2.AND.SR.LT.1.) SR = 1.1
      IF (IALK.EQ.2.AND.ISR.EQ.1) PH=7.0
      IF (IALK.EQ.1.AND.ISR.EQ.2) LG = 50.
      IF (IALK.EQ.1.AND.ISR.EQ.3) PH = 5.5
      IF (IALK.EQ.1.AND.ISCRUB.GE.3) PH=5.5
```

```
      IF (ISR.EQ.3) SO2R = 75.
      IF (IALK.EQ.1.AND.ISCRUB.GE.3.AND.ISR.EQ.3)  SO2RST=0.60
      IF (ISR.EQ.3) XSO2 = 75.
      IF (ISR.EQ.3) ISO2 = 1
      IF (IADD.EQ.2) PH = PHAA
      LSADD=0.0
      LIMEAD=0.0
      SOBOIL = 0.0
      PCTSCR = 0.0
      SCREM = 0.0
      SO2EMT = 0.0
      RMW = MW
      HVCR = HVC
      WPCR = WPC
      WPHR =WPH
      WPOR = WPO
      WPNR = WPN
      WPSULR = WPSUL
      WPCLR = WPCL
      WPASHR = WPASH
      WPH2OR =WPH2O
      ISBCAL=1
C
      IF (IVPD .EQ. 0) GOTO 10000
C     IF VDP IS NOT INPUT THEN CALCULATE VENTURI PRESSURE DROP
C     (SHAWNEE EQUATION)
      VT2 = VT*VT
      VPD = A1*VT2+A2*(VT**A3)*(VLG**A4)
      GO TO 10030
C     CALCULATE VELOCITY
10000 CONTINUE
C     GIVEN VT AS AN IMPLICIT FUNCTION OF VPD, (ABOVE),
C     NEWTON-RAPHSON METHOD IS USED TO SOLVE FOR VT,
C     I.E. SOLVE FOR POSITIVE ROOT OF
C     F(VT) = A1*VT**2 + A2*(VT**A3)*(VLG**A4) - VPD
C     LETTING FIRST TERM OF F(VT) BE A1*VT**A3 YIELDS
C     FIRST APPROXIMATION
      VT1 = (VPD / (A1 + A2 * (VLG **A4))) ** (1. / A3)
      C1 = A2 * (VLG **A4)
      C2 = 2 * A1
      C3 = A3 * C1
      C4 = A3 - 1
      EPS = .0001
      DO 10010 I = 1,50
      VT2 = VT1 - (A1*VT1*VT1 + C1*(VT1**A3) - VPD)
     1           /(C2 * VT1 + C3 * (VT1 **C4))
      IF ( ABS((VT1-VT2)/(VT1+VT2)) .LT. EPS) GOTO 10020
      VT1 = VT2
10010 CONTINUE
      ICOMPL = 1
      WRITE(6,20000)
20000 FORMAT(' VENTURI VELOCITY FAILED TO CONVERGE ')
```

```
      GO TO 10030
10020 CONTINUE
      VT = VT2
10030 CONTINUE
C
      ISBCAL=2
C
      ISBCAL=3
C
      ISBCAL=4
C      CALL DUST
      YSO2D = MPCT(3) / 100.0
      CFR = 1000.*MW*BHR/HVC
      IF(INPOPT.NE.1) GO TO 60400
C
C**** COMPOSITION OF COMBUSTION AIR, WT PCT (DRY BASIS)
      N2AIR = 76.85
      O2AIR = 23.15
      H2OAIR = 1.338
C
C**** CALCULATE COMBUSTION PRODUCTS, LB/100 LB COAL
C
      F(1) = WPC*MDUST(1)/MDUST(7)
      F(2) = WPCL*MDUST(2)/MDUST(8)
      SULOF = SULO/100.
      F(3) = WPSUL*SULOF*MDUST(3)/MDUST(9)
      FO2C = WPC*MDUST(4)/MDUST(7)
      FO2H = (WPH-WPCL*MDUST(10)/MDUST(8))*MDUST(4)/(4.*MDUST(10))
      FO2S = WPSUL*SULOF*MDUST(4)/MDUST(9)
      XSF = EXSAIR/100.
      SO2100 = WPSUL*MDUST(4)/MDUST(9)
C     SO2100 ALL SULFUR IN COAL OXIDIZED , ALSO SUBROUTINE HOTGAS
      F(4) = (FO2C+FO2H+FO2S-WPO)*XSF
      O2R=(1.+XSF)*(FO2C+FO2H+SO2100-WPO)
      F(5) = WPN+(N2AIR/O2AIR)*O2R
      FH2OH = MDUST(6)*(WPH-WPCL*MDUST(10)/MDUST(8))/(2.*MDUST(10))
      FH2OA = (H2OAIR/O2AIR)*O2R
      F(6) = FH2OH+FH2OA+WPH2O
      ASHOF = ASHO/100.
      FAIR = (100./O2AIR)*O2R
      FG = (F(1)/MDUST(1)) + (F(2)/MDUST(2)) + (F(3)/MDUST(3)) +
     +(F(4)/MDUST(4)) + (F(5)
     1 /MDUST(5)) + (F(6)/MDUST(6))
C
C**** CALCULATE HOT GAS COMPOSITION AND FLOW RATE
C
      FGI = 100./FG
      C100 = CFR/100.
      YSO2D =(F(3)*FGI/MDUST(3))/100.
60400 CONTINUE
      SOBOIL=CFR*(WPSUL/100.0)*(SULO/100.0)*64.06/32.06
      IF(INPOPT.EQ.2) SOBOIL = SCFM * YSO2D * 64.04 * 60./ 379.
```

```
      SO2EMT=XSO2*BHR*1000.0*MW/1.0E6
      IF(ISO2.EQ.1) SO2EMT = SOBOIL * (1.0-XSO2/100.0)
      IF(ISO2.NE.3) GO TO 60500
      PPMHOT = YSO2D * 1.0E6
      SO2EMT = SOBOIL * (XSO2/PPMHOT)
60500 CONTINUE
      SCREM=SOBOIL-SO2EMT
      PCTSCR = (SCREM/PSSO2X)/SOBOIL * 10000.0
      PCTBYP=100.0-PCTSCR
      RCFR=CFR
      IF (INPOPT.NE.1) WPASH=1E4*WASH/(CFR*ASHO)
      ASHBTU=(ASHO/100.0)*(WPASH/100.0)*(1E6/HVC)
      ASHPRU=0.0
      ASHPRS=99.2
      IF (IESP.EQ.1) ASHPRU=33.0
      IF (IASH.EQ.0) GO TO 61010
      ASHPRU=ASHUPS
      ASHLBU=ASHUPS
      ASHPRS=ASHSCR
      ASHLBS=ASHSCR
      IF (IASH-2) 61010,61020,61000
61000 CONTINUE
      ASHPRS=99.2
      IF (ASHPRU.GT.97.0) ASHPRS=75.0
61010 CONTINUE
      ASHLBU=ASHBTU*(1.0-ASHPRU/100.0)
      ASHLBS=ASHLBU*(1.0-ASHPRS/100.0)
      GO TO 61040
61020 CONTINUE
      IF(KPASO2.NE.1) GO TO 61021
      ABOILR=CFR*(WPASH/100.0)*(ASHO/100.0)
      AEMITT=ASHSCR*BHR*1000.0*MW/1.0E6
      ASHPRS=100.0*(1.-ASHSCR/ASHUPS)
      DENOM=PCTBYP/100.0+(1.-ASHPRS/100.0)*(1.-PCTBYP/100.0)
      ALBHUS=AEMITT/DENOM
      ASHPRU=100.0*(ABOILR-ALBHUS)/ABOILR
      ASHLBU=ALBHUS*1.0E6/(BHR*1000.0*MW)
      ASHLBS=ASHLBU*(1.0-ASHPRS/100.0)*(1.-PCTBYP/100.0)
      GO TO 61031
61021 ASHPRU=100.0*(1.0-ASHLBU/ASHBTU)
      IF (ASHPRU.LT.0.0) ASHPRU=0.0
      IF (ASHPRU.EQ.0.0) ASHLBU=ASHBTU
      IF (ASHPRU.LT.100.0) GO TO 61030
      ASHPRU=100.0
      ASHPRS=0.0
      ASHLBU=0.0
      ASHLBS=0.0
      GO TO 61040
61030 CONTINUE
      ASHPRS=100.0*(1.0-ASHLBS/ASHLBU)
61031 IF (ASHPRS.LT.0.0) ASHPRS=0.0
      IF (ASHPRS.EQ.0.0) ASHLBS=ASHLBU
```

```
61040 CONTINUE
      IF(KPASO2.EQ.1) ASHUPS=ASHLBU
      IF(KPASO2.EQ.1) ASHSCR=ASHLBS
      PR=ASHPRS
C
      EBYPAS=50.0
      ISBCAL=5
      CALL HOTGAS
C
      IF (ICOMPL.NE.0) RETURN
      ISBCAL=6
C
C     CALCULATE SO2 EMISSION DESIGN
C
      SO2BTU=(WPSUL/100.0)*(SULO/100.0)*(64.06/32.06)*(1E6/HVC)
      GO TO (1002,1011,1031,1011), ISO2
 1002 CONTINUE
      SO2REF=XSO2
      SO2LMB=SO2BTU*(1.0-SO2REF/100.0)
      GO TO 1021
 1011 CONTINUE
      SO2LMB=XSO2
      SO2REF=(1.0-SO2LMB/SO2BTU)*100.0
 1021 CONTINUE
      SO2PPM=YSO2*1E6*(1.0-SO2REF/100.0)*(1.0-YWH2O)/(1.0-YH2O)
      GO TO 1041
 1031 CONTINUE
      SO2PPM=XSO2
      SO2REF=(1.0-SO2PPM*(1.0-YH2O)/((1.0-YWH2O)*YSO2*1E6))*100.0
      SO2LMB=SO2BTU*(1.0-SO2REF/100.0)
 1041 CONTINUE
      SO2R=SO2REF
      ELIMIT=SO2LMB
      PPMOUT=SO2PPM
C
      IF(KPASO2.NE.1) GO TO 1010
      ISBCAL=7
      CALL BYPASS
      EBYPAS=50.0+0.5*PASSBY
 1010 CONTINUE
      IF(NOTRAN.LT.2) EBYPAS=100.0
      ISBCAL=8
C     CALL PRIN
      IF(ICOMPL.NE.0) RETURN
      ISBCAL=9
      KGTA = 3.
      KGEHT = 0.3
      KPCO2 = 0.08
      C1 = 0.0046
      C2 = 1.28
      C3 = 0.10
      C4 = 0.
```

```
      PC1 = 0.221
      PC2 = 0.0355
      PC3 = -1.89
      PC4 = 1.68
      PC5 = 0.0000717
C
      IF(IADD.EQ.2) GO TO 1055
 1020 CONTINUE
      ISBCAL=12
      CALL MATBAL
      IF (IALK.NE.1.AND.ISR.EQ.0) GO TO 1150
      IF (ISR.EQ.0) GO TO 1140
      SROLD = SR
      OLDLG=LG
      PHOLD=PH
      ROLD=SO2R
      IF (ISR.NE.3.OR.IALK.EQ.2) GO TO 1030
      ISBCAL=13
      ICR=2
      CALL SRMOD(ICR)
      ISBCAL=14
      ICR=1
      CALL SRMOD(ICR)
      IF (ICOMPL.NE.0) RETURN
      GO TO 1070
 1030 CONTINUE
      ISBCAL=15
      ICR=1
      CALL SRMOD(ICR)
C
      IF (ICOMPL.NE.0) RETURN
      IF (IALK.EQ.2.AND.ISR.EQ.3) GO TO 1100
      IF (IALK.NE.1) GO TO 1150
      IF (ISR.EQ.2) GO TO 1040
      ISBCAL=16
      ICR=2
      CALL SRMOD(ICR)
      GO TO 1070
 1040 CONTINUE
      IF (IT.GT.ITMAX.AND.ISR.EQ.2) GO TO 1060
      IT = IT+1
      TOLLG = 0.01
C      IF (JIT.NE.1) GO TO 1050
 2000 FORMAT (/1H ,11X,'IT',5X,'LG',7X,'OLDLG')
 2050 FORMAT (1H ,9X,I4,5X,F4.0,5X,F4.0)
 1050 CONTINUE
      IF (ABS((LG-OLDLG)/LG).GT.TOLLG) GO TO 1020
      GO TO 1150
 1060 CONTINUE
      WRITE (6,2100) ITMAX
 2100 FORMAT (/1H ,'L/G CALCULATION UNCONVERGED AFTER',I4,
     1 ' ITERATIONS'/)
```

```
      GO TO 1140
 1070 CONTINUE
      IF (ISR.EQ.2) GO TO 1150
      IF (ISR.EQ.3) GO TO 1100
      IT = IT + 1
      IF (JIT.NE.1)  GO TO 1080
 2150 FORMAT (/1H ,11X,'IT',5X,'SR',7X,'SROLD')
 2200 FORMAT (1H ,9X,I4,5X,F4.2,5X,F4.2)
 1080 CONTINUE
      IF (IT.GT.ITMAX) GO TO 1090
      TOL = 0.002
      IF (ABS((SR-SROLD)/SR).GT.TOL) GO TO 1020
 1090 CONTINUE
 2250 FORMAT (/1H ,'STOICHIOMETRY CALCULATION UNCONVERGED AFTER',
     1 I4,' ITERATIONS'/)
      IF (ISR.NE.3) GO TO 1140
 1100 CONTINUE
      IT = IT+1
      IF (JIT.NE.1) GO TO 1120
      IF (IT.NE.1) GO TO 1110
 2300 FORMAT (/1H ,11X,'IT',6X,'PH',8X,'PHOLD',5X,'SO2R',4X,'ROLD')
 2350 FORMAT(/1H ,11X,'IT',5X,'SO2R',6X,'ROLD')
 1110 CONTINUE
 2400 FORMAT (1H ,9X,I4,5X,F5.2,5X,F5.2,5X,F5.1,5X,F5.1)
 2450 FORMAT (1H ,9X,I4,5X,F5.1,5X,F5.1)
 1120 CONTINUE
      IF (IT.GT.ITMAX) GO TO 1130
      PHTOL = 0.01
      RTOL=0.1
      IF (IALK.EQ.1.AND.ABS(SO2R-ROLD).GT.RTOL) GO TO 1160
      IF (IALK.LE.2.AND.ABS(SO2R-ROLD).GT.RTOL) GO TO 1020
 1130 CONTINUE
 2500 FORMAT (/1H ,'SO2 REMOVAL CALCULATION UNCONVERGED AFTER',I4,
     1 ' ITERATIONS'/)
 1140 CONTINUE
      ISBCAL=17
      ICR=1
      CALL SRMOD(ICR)
      IF (ICOMPL.NE.0) RETURN
 1150 CONTINUE
C     IF (ISR.EQ.3) PPMOUT=YSO2*1E4*(100.0-SO2R)*(1.0-YWH2O)/(1.0-YH2O)
      GO TO 1056
 1055 CONTINUE
      CALL MATBAL
      CALL ADIPID
      CALL MATBAL
 1056 CONTINUE
      ISBCAL=18
      IF (ICLAR.NE.1) GO TO 10001
C
      F1 = (100.-WPS)/WPS
      D = (100.-PSC)/PSC
```

```
      IF (RS.NE.0.0) GO TO 101
      IF (OX.LE.65.0) RS = 1./(0.744014-(0.0050416*OX))
      IF (OX.GT.65.0) RS = 1./(1.01762-(0.0094793*OX))
  101 CONTINUE
      UA = 1.3333*1.25*(F1-D)/RS
      TPD = WDIS*(PSD/100.)*24./2000.
      CCSA = UA*TPD
10001 CONTINUE
C
      ISBCAL=19
      CALL WETGAS
C
      ISBCAL=20
C
      ACS = WSCFM*(TWG+460.)*(14.696/PRES)/(520.*60.*V)
      ACSTRN=ACS/FLOAT(NOTRAN)
      IF (ACSTRN.LE.1370.0) GO TO 1003
      WRITE(6,2003)
 2003 FORMAT('0','MAXIMUM SCRUBBER SIZE EXCEEDED, NUMBER OF',
     1 ' TRAINS SET EQUAL TO MINIMUM NUMBER')
      NOTRAN=ACS/1370.0+0.999999
      ACSTRN=ACS/FLOAT(NOTRAN)
 1003 CONTINUE
      NTOTAL=NOTRAN+NOREDN
C      CALL CSA
C
      ISBCAL=21
C
      IFLOOD=0
      XNG = NGRID
      XNS = NSTAGE
      L = LG*V*60./1000.
      IF (HS.GT.EPS.AND.NSTAGE.GT.0) GO TO 1004
      PD=PC1*V+PC2*L+PC3
      GO TO 1012
 1004 CONTINUE
      PD = PC4+PC5*HS*XNS*L*V*V
      PDMAX = 3.*XNS
      IF (PD.LE.PDMAX) GO TO 1012
      LMAX = (PDMAX-PC4)/(PC5*HS*XNS*V*V)
      LGMAX = LMAX*1000./(V*60.)
C     PD = PDMAX
      L = LGMAX*V*60./1000.
      PDMAX = PC4+PC5*HS*XNS*L*V*V
      IFLOOD = 1
 1012 CONTINUE
C
      ISBCAL=22
C
      ISBCAL=23
      IF (IRH.EQ.1) GOTO 7770
      DO 71000 I=1,6
```

```
      OD1(I) = OD(I)
71000 CONTINUE
      PDV = 0.0
      IF(ISCRUB.GE.3) PDV = VPD
      PI = 3.1416
      GLBH = 0.0
      GLBH1 = 0.0
      GLBH2 = 0.0
      DO 71010 I=1,6
      ARH(I)= 0.0
      NT(I) = 0.0
      NB(I) = 0.0
71010 CONTINUE
      EN = NTRAIN * 8.33 * 60.
      R = 0.00968
      J = 1
      RHT = 150
C
C        DETERMINE THE HEAT OR COOLING REQUIREMENT NECESSARY TO CHANGE
C        THE TOTAL FLOWRATE OF THE BY-PASSED OR THE WET GAS BY ONE
C        DEGREE F., BTU/HR DEG. F
C
      DO 71020 I = 1,6
      GLBH1 = WLBH(I) * CPS(I) + GLBH1
      GLBH2 = BYLBH(I) * CPH(I) + GLBH2
      GLBH = WLBH(I) + GLBH
71020 CONTINUE
      GLBH1 = GLBH1 + EN*CPS(6)
      GLBH = (GLBH + EN) / NOTRAN
C
C        DETERMINE THE LATENT HEAT OF VAPORIZATION OF WATER AT THE
C        TEMPERATURE OF THE WET GAS FROM THE SCRUBBER, BTU/LB
C
      ELIQUD = 83.97 + (TWG-116.)
      EVAPOR = 1126.2 + (0.4*(TSK-150.))
      HVH2O = EVAPOR - ELIQUD
C
C        DETERMINE THE RESULTANT TEMPERATURE IF THE WET GAS FROM THE
C        SCRUBBER IS MIXED WITH THE BY-PASSED GAS WITHOUT ANY REHEAT
C
      TSKA=TSK-(GLBH2*(THG-TSK)-EN*HVH2O)/GLBH1
      IF(TSKA.LT.TWG) TSKA=TWG
      TEMPS=TSKA
C
C        IF AN I.D. FAN IS TO BE USED THEN DETERMINE THE PRESSURE
C        DROP THROUGH THE SCRUBBER SECTION, INCHES OF WATER
C
      IF(IFAN.NE.1) GO TO 71030
      PDI = PRES * 27.681
      DELTPD = (PD+PDV +3.5) * 1.15
      IF(DELTAP.NE.0.0) DELTPD = DELTAP
C
```

```
C         I.D. FAN COMPRESSION TEMPERATURE RISE PERRY 6-24, 1975
C
      TEMPS = ((TSKA + 460.)/(PDI/(PDI-DELTPD))**0.2857) - 460.
C
CC        IF NO REHEATER IS TO BE DESIGNED THEN TERMINATE EXECUTION OF
CC        THE SUBROUTINE
C
71030 IF(IRH.EQ.0) GO TO 71100
C
CC        DETERMINE THE HEAT SUPPLIED BY THE I.D. FAN, IF APPLICABLE
C
      QFAN=GLBH1*(TSKA-TEMPS)
C
CC        DETERMINE THE HEAT LOAD REQUIRED BY THE REHEATER, IF LESS
CC        THAN ZERO, THEN NO REHEATER IS REQUIRED
C
      Q1=GLBH1*(TSK-TWG)+EN*HVH2O-GLBH2*(THG-TSK)-QFAN
      IF(Q1.LE.0.0) GO TO 71100
C
CC        DETERMINE THE RESULTANT REHEATER OUTLET TEMPERATURE, IF LESS
CC        THAN THE WET GAS TEMP. THEN ONLY PARTIAL VAPORIZATION OF THE
CC        ENTRAINED WATER IS NECESSARY
C
      TEMPS=TWG+(Q1-EN*HVH2O)/GLBH1
      IF(TEMPS.LT.TWG) TEMPS=TWG
      IF(TEMPS.LE.150.0) RHT=TEMPS
      GO TO 71060
C
CC        DETERMINE THE AREA, NO. OF TUBES, AND NO. OF BANKS OF TUBES
C         OF THE INCONEL SECTION OF THE REHEATER
C
71040 DO 71050 I=1,6
      ARINCL(I) = ARH(I)
      NINCLT(I) = NT(I)
      NINCLB(I) = NB(I)
71050 CONTINUE
C
CC        IF REHEATER OUTLET TEMPERATURE IS LESS THAN 150 DEG. F.
CC        THEN DESIGN IS COMPLETE, OTHERWISE DESIGN THE CORTEN SECTION
C
      IF (TEMPS.LE.150) GO TO 71080
      RHT = TEMPS
      J = J+1
C
CC        DETERMINE ALL PHYSICAL CHARACTERISTICS OF THE FLUIDS THAT
C         ARE NEEDED TO DESIGN THE REHEATER
C
71060 DELT = RHT-TWG
      DELT1 = TSTEAM-TWG
      DELT2 = TSTEAM-RHT
      TLN=DLOG(DELT1/DELT2)
      IF((DABS(TLN)).GT.1.0E-7) DTLM=(DELT1-DELT2)/TLN
```

```
      IF((DABS(TLN)).LE.1.0E-7) DTLM=(DELT1+DELT2)/2.0
      IF(DTLM.GT.(DELT1+DELT2).OR.DTLM.LE.0.0) DTLM=(DELT1+DELT2)/2.0
      KM = 0.0208
      TF = (TSTEAM+0.5*RHT+0.5*TWG)/2.
      MUF = 0.0467+0.0055*((TF-100.)/100.)
      TBAR = (TWG+RHT)/2.
      RHOG = 29.*PRES*520./(379.5*14.696*(TBAR+460.))
C
C         DETERMINE THE HEAT LOAD OF THE REHEATER, BTU/HR
C
      Q=Q1
      IF(RHT.LT.TEMPS.AND.RHT.GT.TWG)Q=GLBH1*DELT+EN*HVH2O
      WSTEAM = Q/HVS
      SCS = WACFM/(60.*VRH*NOTRAN)
      PL = SQRT(SCS)
      QPRIME = Q/(SCS*NOTRAN)
      DO 71070 I=1,6
      RCS = SCS/FACTOR(I)
      GMAX = GLBH/RCS
C
C         DETERMINE THE FILM COEFFICIENT (OUTSIDE) AND THE OVERALL
C         HEAT TRANSFER COEFFICIENT FOR THE REHEATER TUBING
C
      H0= 0.811*(KM/OD(I))*((OD(I)*GMAX)/MUF)**0.6
      URH(I) = 1./(R+1./H0)
      APRIME = QPRIME/(URH(I)*DTLM)
C
C         DETERMINE THE TOTAL AREA, NO. OF TUBES, AND NO. OF BANKS OF
C         TUBES IN THE REHEATER
C
      ARH(I) = APRIME*SCS
      XNB = APRIME*12./(NFT(I)*PI*OD(I))
      XNT = ARH(I)*12./(OD(I)*PL*PI*XNB)
      NB(I) = INT(XNB+1.0)
      NT(I) = INT(XNT+1.0)
      N1 = 0.43+1.13/XL(I)
      F1 = (0.044+0.08*XL(I)/((XT(I)-1.)**N1))*((MUF*12./(OD(I)*GMAX))
     1 **0.15)
C
C         DETERMINE THE REHEATER PRESSURE DROP, INCHES OF WATER
C
      DPRH(I) = 48.*F1*XNB*GMAX*GMAX/(64.4*RHOG*62.4*3600.*3600.)
71070 CONTINUE
      IF (J.EQ.1) GO TO 71040
C
C         THE AREA, NUMBER OF TUBES, AND NUMBER OF BANKS OF TUBES IN
C         THE CORTEN SECTION OF THE REHEATER, IF APPLICABLE
C
71080 DO 71090 I=1,6
      ARCORT(I) = ARH(I)-ARINCL(I)
      NCORTT(I) = NT(I)
      NCORTB(I) = NB(I)-NINCLB(I)
```

```
71090 CONTINUE
      IF(IFAN.EQ.1) GO TO 71110
      RETURN
71100 CONTINUE
      TSKA = TWG
      IF(IFAN.EQ.1)TSKA=(TWG+460.)*(PDI/(PDI-DELTPD))**0.2857-460.
C
C       THE RESULTANT STACK GAS TEMPERATURE, IF NO REHEATER IS USED
C
      TSK = (GLBH2*THG+GLBH1*TSKA-EN*HVH2O)/(GLBH1+GLBH2)
      IRH = 0
71110 CONTINUE
C
C     I.D. FAN TEMP CORRECTION
C
      IF(TWG.GE.TEMPS) TEMPS = TWG
      TGAS = TEMPS - TWG
      TGASID = TSKA - TEMPS
 7770 CONTINUE
C
      ISBCAL=24
      IF (IRH.EQ.1)GO TO 1016
      RWDUST=0
      RHSCFM=0
      DO 1006 I=1,6
      RHLBH(I)=0
 1006 CONTINUE
 1016 CONTINUE
      EN = NTRAIN*8.33*60.
      DO 1026 I=1,5
      SLBH(I) = RHLBH(I) + WLBH(I) + BYLBH(I)
      SLBMH(I) = SLBH(I)/M(I)
 1026 CONTINUE
      SLBH(6) = RHLBH(6) + WLBH(6) +BYLBH(6) + EN
      SLBMH(6) = SLBH(6)/M(6)
      SUM = SLBMH(1)+SLBMH(2)+SLBMH(3)+SLBMH(4)+SLBMH(5)+SLBMH(6)
      DO 1036 I=1,6
      SMPCT(I) = SLBMH(I)*100./SUM
 1036 CONTINUE
      SSCFM = RHSCFM + WSCFM + BSCFM + (EN*379.)/(60.*M(6))
      SACFM = SSCFM*(14.696/PRES)*(TSK+460.)/520.
      BYDUST = WASH * PASSBY/100.
      SWDUST = RWDUST + WASHOT + BYDUST
      SDUST = SWDUST*7000./(60.*SSCFM)
C
      ISBCAL=25
C
      ISBCAL=26
      CALL STREAM
C
      ISBCAL=27
C
```

```
      MH2O = 18.016
      MCASO3 = 129.1
      MCASO4 = 172.1
      IF (IALK.EQ.1) WLSM = LSADD*(WPM/100.)/(8.33*60.)
      IF (IALK.EQ.2) WLSM = LIMEAD*(WPM/100.)/(8.33*60.)
      H2OHYD = MH2O*((CASO3*0.5/MCASO3)+(CASO4*2./MCASO4))/(8.33*60.)
      IF (ISLUDG.LT.3) N=4
      IF (ISLUDG.EQ.3) N=13
      IF (ISLUDG.GE.4) N=16
      TWR=WVAPOR+NTRAIN+WFLOW(N,5)/(8.33*60.0)+H2OHYD+EVAP+EVAPC+SEEP
      TWA=RAFAL+WLSM+RAFALC
      MUW=TWR-TWA
      GPM (5) = MUW
C
      ISBCAL=28
      FILTSA=0.0
      IF (ISLUDG.LT.4) GO TO 1001
      FILTSA=WDIS/2000.0*PSD/100.0*24.0/FILRAT
 1001 CONTINUE
C
      GO TO 1170
 1160 CONTINUE
      SO2R = (SO2R+ROLD)/2.
      GO TO 1020
 1170 CONTINUE
 1180 CONTINUE
      RETURN
      END
```

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **
C
C     SUBROUTINE ADIPID    SCRUB PARAMETERS  ADIPIC ACID ADDITION
C
C     REVISION DATE 82-04-08
C
C * ** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE ADIPID
C
      COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
     1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
     2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
     3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
     4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
     5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
     6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
      COMMON/COMPLE/ICOMPL, IRPT, IRUN, NEXT, IEQPR, IWTBAL, ITM, INPOPT, NOPART
      COMMON/EMIT/ISO2, XSO2, ISSO2, ELISO2, SO2REF, SO2LMB, SO2PPM
      COMMON/HGAS/ACFM, THG, YCO2, YHCL, YSO2, YO2, YN2, YH2O, ASH, WCL, WASH, IESP
      COMMON/LIQDIS/H2O, CAPPM, MGPPM, SO3PPM, SO4PPM, CLPPM, TDS, PHPOND, DGPM
      COMMON/LIQ2/IW4, AD, ADDC, ADLOSS, AMGLOS
      REAL MGPPM
      COMMON/LS/IALK, ISR, SR, SRADD, WPMGO, WPI, WPM, LSADD, LIMEAD, MGOADD, WMG
      REAL LSADD, LIMEAD, MGOADD
      COMMON/OUTPT/JOUTPT, JIT, JEQ1, JEQ2, JEQ3
      COMMON/RTIME/TR, ALCONT(9)
      COMMON/SCRUB/ISCRUB, IFOX, IADD, VLG, VTR, IFAN
      COMMON/SCRUBR/LG, V, PPMOUT, SO2R, ELIMIT, PR, PD, WPS, OX, PH, NSTAGE,
     1       NGRID, HS, ACS, ACSTRN
      REAL LG
C
      REAL NTANKS, MGCLQ, L
      REAL NTUST, NTUV
C
      IF(ISR.EQ.0) GO TO 450
      IF(ISCRUB.GT.1) GO TO 30
      IF(OX.GT.50.0) GO TO 20
C    SPRAY TOWER LOW OXIDATION
      B1 = -4.5E-5
      B2 = 0.86
      B3 = 1.2
      B4 = 5.0E-4
      GO TO 50
C    SPRAY TOWER HIGH OXIDATION
   20 B1 = -2.5E-3
      B2 = 0.92
      B3 = 0.47
      B4 = 4.0E-4
      GO TO 50
C    TCA  HIGH PH
```

```
   30 IF(ISCRUB.GT.2) GO TO 40
      C1 = -0.0645
      C2 = 0.76
      C3 = 0.038
      C4 = 0.0177
      C5 = 0.5
      C6 = 1./C2
      L = LG*V*60./1000.
      FHS = HS * NSTAGE
      IF(ISR.NE.2) GO TO 650
      AL = (ALOG(1-SO2R/100) / EXP((C3*FHS)+C4*(AD**C5)))/C1
      ALL = AL **C6
      LG = ALL*1000./(V*60.)
      GO TO 400
  650 IF(ISR.EQ.3)SO2R=100.*(1-EXP((C1*L**C2)*EXP(C3*FHS+C4*(AD**C5))))
      IF(ISR.EQ.3) GO TO 400
      AA1 = (ALOG(1-SO2R/100.)/(C1*L**C2))
      AA2 = (ALOG(AA1) - FHS*C3)/C4
      AD = AA2**2
      GO TO 400
C     VENTURI SPRAY TOWER LIMESTONE
   40 B1 = -0.0019
      B2 = 0.55
      B3 = 0.8
      B4 = 8.0E-4
      LG = LG + VLG - 21.
   50 CONTINUE
      F1 = ALOG(100/(100-SO2R))
      GO TO (400,200,300,300),ISR
  200 ALG = ((ALOG(1-SO2R/100))/EXP((B3*PH)+B4*AD))/B1
      B5 = 1./B2
      LG = ALG**B5
      GO TO 400
  300 IF(ISR.EQ.3)SO2R=100*(1-EXP((B1*(LG)**B2)*EXP(B3*PH+B4*AD)))
      IF(ISR.EQ.3) GO TO 400
      AA1 = (ALOG(1-SO2R/100)/(B1*LG**B2))
      AD =(ALOG(AA1) -(PH*B3))/B4
  400 CONTINUE
      IF(ISCRUB.EQ.2) GO TO 450
      PH1 = (ALOG(1-SO2R/100)/(B1*(LG)**B2))
      PH = (ALOG(PH1)-B4*AD)/B3
      IF(ISCRUB.GE.3) LG = LG-VLG+21.
  450 CONTINUE
      RETURN
      END
```

```
$NOFLOATCALLS
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM  BYPASS  TO DESIGN FOR PARTIAL SCRUBBING
C     REVISION DATE 81-03-11
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE BYPASS
C
      COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
     1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
     2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
     3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
     4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
     5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
     6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
      COMMON/BOILER/MW, BHR, EXSAIR, RMW, PREC, SPASH, WPRITE, KCLEAN, SMRW, SMCL
      REAL MW
      COMMON/BYPAS2/BYLBH(6), BYLBMH(6), BSCFM, BACFM, PASSBY, XPASSD, IBY
      COMMON/BYPAS3/BOLBH(6), BOLBMH(6), BOSCFM, BOACFM
      COMMON/COAL/HVC, WPC, WPH, WPO, WPN, WPSUL, WPCL, WPASH, WPH2O, SULO, ASHO
      COMMON/COMPLE/ICOMPL, IRPT, IRUN, NEXT, IEQPR, IWTBAL, ITM, INPOPT, NOPART
      COMMON/EMIT/ISO2, XSO2, ISSO2, ELISO2, SO2REF, SO2LMB, SO2PPM
      COMMON/HGAS/ACFM, THG, YCO2, YHCL, YSO2, YO2, YN2, YH2O, ASH, WCL, WASH, IESP
      COMMON/HGAS2/MPCT, LBMH, LBH, SCFM, HHG, HWG, CFR, BCFR, RCFR
      REAL MPCT(6), LBMH(6), LBH(6)
      COMMON/RCOAL/HVCR, WPCR, WPHR, WPOR, WPNR, WPSULR, WPCLR, WPASHR, WPH2OR
      COMMON/WGCOMP/YWCO2, YWHCL, YWSO2, YWO2, YWN2, YWH2O, WVAPOR, NTRAIN
      REAL NTRAIN
      COMMON/SCRUBR/LG, V, PPMOUT, SO2R, ELIMIT, PR, PD, WPS, OX, PH, NSTAGE,
     &      NGRID, HS, ACS, ACSTRN
C
      BBTU = MW * BHR/1000.
      SO2CLN = ((BBTU*WPSUL)/HVC)*SULO*1.E2*64.06/32.06
      IF(ISO2.EQ.3) GO TO 1000
C     PBTU = PREC/100.*HVC/HVCR
C     CMBTU = BBTU/PBTU
C     SULLBS = 1.E4 * WPSULR / HVCR
C     SO2CN2 = CMBTU * SULLBS * (64.06/32.06)
C     EMIS = SO2CN2 *  (100 - SO2REF)/100
C     IF(EMIS.LT.SO2EMS) SO2EMS = EMIS
      SO2EMS = BBTU * SO2LMB
      PASSUP = (SO2EMS-SO2CLN*(1.0-PSSO2X/100.0))/(PSSO2X/100.0)
      PASSBY = (PASSUP/SO2CLN) * 100.
      GO TO 1010
 1000 CONTINUE
      PPMBYP=YSO2*1E6
      PPMSCR=PPMBYP*(1.0-PSSO2X/100.0)*((1.0-YWH2O)/(1.0-YH2O))
      PPMSTK=XSO2
      PASSBY=100.0*(PPMSTK-PPMSCR)/(PPMBYP-PPMSCR)
```

```
C     CORRECTION FOR WATER PICKED UP IN SCRUBBER
      PASSBY=100.0/(1.0+(100.0/PASSBY-1.0)*((1.0-YWH2O)/(1.0-YH2O)))
      SO2EMS=SO2CLN*(1.0-(1.0-PASSBY)*PSSO2X/100.0)
 1010 CONTINUE
      IF(PASSBY.LT.0.1) GO TO 1030
      GASCRB = 100. - PASSBY
      IF(PASSBY.GE.99.5) GO TO 1020
      MW = (GASCRB * MW) / 100.
C
C     SAVE HOT GAS FROM BOILER AND CALCULATE TO BYPASS AND SCRUBBERS
C
      XBY=PASSBY/100.0
      XSC=1.0-XBY
      RCFR=CFR
      BCFR=XBY*CFR
      CFR=XSC*CFR
      DO 1015 I=1,6
      BOLBH(I)=LBH(I)
      BOLBMH(I)=LBMH(I)
      BYLBH(I)=XBY*LBH(I)
      BYLBMH(I)=XBY*LBMH(I)
      LBH(I)=XSC*LBH(I)
      LBMH(I)=XSC*LBMH(I)
 1015 CONTINUE
      BOACFM=ACFM
      BOSCFM=SCFM
      BACFM=XBY*ACFM
      BSCFM=XBY*SCFM
      ACFM=XSC*ACFM
      SCFM=XSC*SCFM
      ISO2=1
      XSO2=PSSO2X
C
C
C     CALCULATE SO2 EMISSION DESIGN
C
      SO2BTU=(WPSUL/100.0)*(SULO/100.0)*(64.06/32.06)*(1E6/HVC)
      GO TO (1001,1011,1031,1011), ISO2
 1001 CONTINUE
      SO2REF=XSO2
      SO2LMB=SO2BTU*(1.0-SO2REF/100.0)
      GO TO 1021
 1011 CONTINUE
      SO2LMB=XSO2
      SO2REF=(1.0-SO2LMB/SO2BTU)*100.0
 1021 CONTINUE
      SO2PPM=YSO2*1E6*(1.0-SO2REF/100.0)*(1.0-YWH2O)/(1.0-YH2O)
      GO TO 1041
 1031 CONTINUE
      SO2PPM=XSO2
      SO2REF=(1.0-SO2PPM*(1.0-YH2O)/((1.0-YWH2O)*YSO2*1E6))*100.0
      SO2LMB=SO2BTU*(1.0-SO2REF/100.0)
```

```
 1041 CONTINUE
      SO2R=SO2REF
      ELIMIT=SO2LMB
      PPMOUT=SO2PPM
      GO TO 1040
C
 1020 CONTINUE
      WRITE(6,2050)
 2050 FORMAT(1H,9X,'NOT NECESSARY TO SCRUB, CASE TERMINATED'/)
      ICOMPL = 1
      IZEROC = 0
 2173 FORMAT(3I3)
      GO TO 1040
 1030 CONTINUE
      PASSBY = 0.0
      IBY = 1
 2100 FORMAT(1H,9X,'ZERO BYPASS ALLOWED BY INPUT CONDITIONS')
 1040 CONTINUE
      RETURN
      END
```

```
$NOFLOATCALLS
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C      SUBPROGRAM HOTGAS
C      REVISION DATE 82-06-18
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       SUBROUTINE HOTGAS
C
       COMMON /OPTIM/BYPSO2, BYPMST, BYPCFM
       COMMON /OPT/     BYPFGD, BYPFF, EFFESP, EFFLSD, EFFLIM
       COMMON /STREAM/ PPH, ACFM1, TEMP, PPHMST, PPHALK, PPHPRT, PPHSO2, PPHNOX
       COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
      1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
      2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
      3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
      4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
      5 ISLUDG, IFIXS, JSR, IVPD, IIIT, LON, LGMAX, ISTAND, NSBR,
      6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
       COMMON/BOILER/MW, BHR, EXSAIR, RMW, PREC, SPASH, WPRITE, KCLEAN, SMRW, SMCL
       REAL MW
       COMMON /NEW/ IFACT, VC, FC
       COMMON/BYPAS2/BYLBH(6), BYLBMH(6), BSCFM, BACFM, PASSBY, XPASSD, IBY
       COMMON/BYPAS3/BOLBH(6), BOLBMH(6), BOSCFM, BOACFM
       COMMON/COAL/HVC, WPC, WPH, WPO, WPN, WPSUL, WPCL, WPASH, WPH2O, SULO, ASHO
       COMMON/COMPLE/ICOMPL, IRPT, IRUN, NEXT, IEQPR, IWTBAL, ITM, INPOPT, NOPART
       COMMON/FLYASH/IASH, ASHUPS, ASHSCR, ASHPRU, ASHPRS, ASHLBU, ASHLBS
       COMMON/HGAS/ACFM, THG, YCO2, YHCL, YSO2, YO2, YN2, YH2O, ASH, WCL, WASH, IESP
       COMMON/HGAS2/MPCT, LBMH, LBH, SCFM, HHG, HWG, CFR, BCFR, RCFR
       REAL MPCT(6), LBMH(6), LBH(6)
       COMMON/WGAS/WACFM, WSCFM, TWG, ASHOUT, WASHOT, WMPCT, WLBMH, WLBH
       REAL WMPCT(6), WLBMH(6), WLBH(6)
       COMMON/WGCOMP/YWCO2, YWHCL, YWSO2, YWO2, YWN2, YWH2O, WVAPOR, NTRAIN
       REAL NTRAIN
C
       REAL M(10), MFR(6), F(6), N2AIR, MWDG
       DATA M/44.01, 36.46, 64.06, 32., 28.02, 18.016, 12.01, 35.45, 32.06, 1.008/
       DATA ITMAX, TTOL/20, 0.5/
C      REAL*8 CMP(6)/'CO2', 'HCL', 'SO2', 'O2 ', 'N2 ', 'H2O'/
C
C
C**** COMPOSITION OF COMBUSTION AIR, WT PCT (DRY BASIS)
       N2AIR = 76.85
       O2AIR = 23.15
       H2OAIR = 1.338
C
C**** CALCULATE COMBUSTION PRODUCTS, LB/100 LB COAL
C
       IF (INPOPT.NE.1) GO TO 1010
       F(1) = WPC*M(1)/M(7)
```

```
      F(2) = WPCL*M(2)/M(8)
      SULOF = SULO/100.
      F(3) = WPSUL*SULOF*M(3)/M(9)
      FO2C = WPC*M(4)/M(7)
      FO2H = (WPH-WPCL*M(10)/M(8))*M(4)/(4.*M(10))
      FO2S = WPSUL*SULOF*M(4)/M(9)
      XSF = EXSAIR/100.
      SO2100 = WPSUL*M(4)/M(9)
C     SO2100 ALL SULFUR IN COAL 100% OXIDIZED, SEE SUBROUTINE DUST
      F(4) = (FO2C+FO2H+FO2S-WPO)*XSF
      O2R=(1.+XSF)*(FO2C+FO2H+SO2100-WPO)
      F(5) = WPN+(N2AIR/O2AIR)*O2R
      FH2OH = M(6)*(WPH-WPCL*M(10)/M(8))/(2.*M(10))
      FH2OA = (H2OAIR/O2AIR)*O2R
      F(6) = FH2OH+FH2OA+WPH2O
      ASHOF = ASHO/100.
      FASH=WPASH*ASHOF*(1.0-ASHPRU/100.0)
      FAIR = (100./O2AIR)*O2R
      CFR = 1000.*MW*BHR/HVC
      C100 = CFR/100.
      FG = (F(1)/M(1)) + (F(2)/M(2)) + (F(3)/M(3)) + (F(4)/M(4)) + (F(5)
     1 /M(5)) + (F(6)/M(6))
C
C**** CALCULATE HOT GAS COMPOSITION AND FLOW RATE
C
      FGI = 100./FG
      SCFM = 379.*CFR/(60.*FGI)
      WASH = FASH*C100
      DO 1007 I=1,6
      MPCT(I) = F(I)*FGI/M(I)
      LBMH(I) = F(I)*C100*(1.-BYPFGD)/M(I)
      IF(I .EQ. 3 .OR. I .EQ. 6) LBMH(I)=LBMH(I)/(1.-BYPFGD)
      LBH(I) = LBMH(I)*M(I)
 1007 CONTINUE
      BYPSO2=(LBH(3)/M(3))*BYPFGD
      LBMH(3)=LBH(3)/M(3)-BYPSO2
      LBH(3)=LBMH(3)*M(3)
      BYPMST=(LBH(6)/M(6))*BYPFGD
      LBMH(6)=LBH(6)/M(6)-BYPMST
      LBH(6)=LBMH(6)*M(6)
      GO TO 1030
 1010 CONTINUE
      CFR = 1000.*MW*BHR/HVC
      C100 = CFR/100.
      FG=(SCFM/CFR)*(60.0*100.0/379.0)
      FGI = 100./FG
      DO 1020 I=1,6
      F(I)=MPCT(I)*M(I)*FG/100.0
      LBMH(I)=MPCT(I)*SCFM*60.0/(379.0*100.0)
      LBH(I)=LBMH(I)*M(I)
 1020 CONTINUE
      WPC=F(1)*M(7)/M(1)
```

```
      WPCL=F(2)*M(8)/M(2)
      WPN=0.0
      WPO=0.0
      WPH=0.0
      WPH2O=0.0
      WPSUL=MPCT(3)*FG*M(9)/SULO
      WPASH=1E4*WASH/(CFR*ASHO)
      WASH=WASH*(1.0-ASHPRU/100.0)
      SULOF=SULO/100.0
      ASHOF=ASHO/100.0
      FASH=WPASH/C100
 1030 CONTINUE
      ASH = 7000.*FASH/(379.*FG)
      FBASH =WPASH*(1.-ASHOF)+WPSUL*(1.-SULOF)
      ACFM = ACFM1
      YCO2 = MPCT(1)/100.
      YHCL = MPCT(2)/100.
      YSO2 = MPCT(3)/100.
      YO2 = MPCT(4)/100.
      YN2 = MPCT(5)/100.
      YH2O = MPCT(6)/100.
      WCL = LBH(2)*M(8)/M(2)
C
C**** CALCULATE WET BULB TEMPERATURE AND WET GAS MOISTURE CONTENT
C
      MWDG= (F(1)+F(2)+F(3)+F(4)+F(5))/(FG-F(6)/M(6))
      HHG = YH2O*M(6)/(MWDG*(1.-YH2O))
      HHGMOL=YH2O/(1.-YH2O)
      CP=7.0+2.55*YCO2+8.46*HHGMOL
      IT=0
      TWG=125.
 1040 CONTINUE
      IT=IT+1
      TWX=TWG
      DH=CP*(THG-TWG)/(19692.-10.3*TWG)
      PW=PRES*(HHGMOL+DH)/(1.+HHGMOL+DH)
      TWG=(ALOG(PW)+2.72)/0.027
      IF (IT.LE.ITMAX) GO TO 1050
 2000 FORMAT(1H ,5X,'WET GAS TEMP UNCONVERGED AFTER ',I3,
     1    ' ITERATIONS.'/1H ,5X,'**** CASE TERMINATED.  ****'///)
      ICOMPL=1
      RETURN
 1050 CONTINUE
      IF (ABS(TWG-TWX).GT.TTOL) GO TO 1040
      YWH2O=PW/PRES
      HWG=YWH2O*M(6)/(MWDG*(1.-YWH2O))
      WRITE(8,*) ' LBH,MOLE PERCENT,VC,FC =',LBH,MPCT,VC,FC
      RETURN
      END
```

```
$NOFLOATCALLS
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM MATBAL
C     REVISION DATE 83-08-30
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE MATBAL
C
      COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
     1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
     2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
     3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
     4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
     5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
     6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
      COMMON/ALKASH/ASHCAO, ASHMGO, SRCAO, SRMGO, SRTOT
      COMMON/BYPAS2/BYLBH(6), BYLBMH(6), BSCFM, BACFM, PASSBY, XPASSD, IBY
      COMMON/HGAS/ACFM, THG, YCO2, YHCL, YSO2, YO2, YN2, YH2O, ASH, WCL, WASH, IESP
      COMMON/HGAS2/MPCT, LBMH, LBH, SCFM, HHG, HWG, CFR, BCFR, RCFR
      REAL MPCT(6), LBMH(6), LBH(6)
      COMMON/LIQDIS/H2O, CAPPM, MGPPM, SO3PPM, SO4PPM, CLPPM, TDS, PHPOND, DGPM
      REAL MGPPM
      COMMON/LIQ2/IW4, AD, ADDC, ADLOSS, AMGLOS
      COMMON/LS/IALK, ISR, SR, SRADD, WPMGO, WPI, WPM, LSADD, LIMEAD, MGOADD, WMG
      REAL LSADD, LIMEAD, MGOADD
      COMMON/OUTPT/JOUTPT, JIT, JEQ1, JEQ2, JEQ3
      COMMON/SCRUBR/LG, V, PPMOUT, SO2R, ELIMIT, PR, PD, WPS, OX, PH, NSTAGE,
     1       NGRID, HS, ACS, ACSTRN
      REAL LG
      COMMON/SOLDIS/CASO3, CASO4, CACO3, INSOLS, WDIS, PSD, DENS, DEND
      REAL INSOLS
      COMMON/WGAS/WACFM, WSCFM, TWG, ASHOUT, WASHOT, WMPCT, WLBMH, WLBH
      REAL WMPCT(6), WLBMH(6), WLBH(6)
      COMMON/WGCOMP/YWCO2, YWHCL, YWSO2, YWO2, YWN2, YWH2O, WVAPOR, NTRAIN
      REAL NTRAIN
C
      REAL INERTS, LIQUID, MG, MGLIQM, MGLIQ, MGCONC, MGADD, LSCA
      REAL MCA, MCAO, MMG, MMGO, MSO2, MSO3, MSO4, MCO3, MCL, MH2O
      REAL MCACO3, MCASO3, MCASO4
      DIMENSION X(50)
      DATA MH2O, MCA, MCAO, MMG, MMGO, MCL/18.016, 40.08, 56.08, 24.31, 40.31,
     1     35.45/
      DATA MCO3, MCACO3/60.01, 100.09/
      DATA MSO2, MSO3, MSO4, MCASO3, MCASO4/64.06, 80.06, 96.06, 129.15, 172.15/
C
      EPS = 1.E-8
      FOX = OX/100.
      FI = WPI/100.
      TREAT = (100.-PASSBY)/100.
```

```
      ASHFI = 1. - (ASHCAO/100.) - (ASHMGO/100.)
      ASHREM = WASH*(PR/100.)
      FMGO = WPMGO/100.
      FMG = FMGO*MMG/MMGO
      WPLS = 100.-WPMGO-WPI
      DH = HWG-HHG
      CFMH2O = DH*(LBH(1)+LBH(2)+LBH(3)+LBH(4)+LBH(5))*379./(MH2O*60.)
      YWSO2 = YSO2*SCFM/(SCFM+CFMH2O)
      IF (ISR.EQ.3) PPMOUT=(100.0-SO2R)*YWSO2*1E4
      Y2 = PPMOUT*1.E-6
      FSO2R = 1. - (Y2/YWSO2)
      SO2R = FSO2R*100.
C      PREMSIS CHLORIDE REMOVAL IN SCRUBBER 95%
      WRITE(8,*) ' FSO2,LBH(3),Y2',FSO2,LBH(3),Y2
      SO2 = FSO2R*LBH(3)
      ACID = (SO2/MSO2 + (0.95*TREAT*LBMH(2))/2.)
      OXAIR = SRAIR*SO2*(16./MSO2)/0.2315
      POX = OX
      IF(OX.LE.15.) POX = 15.
      OXAIR = OXAIR * (POX-15.)/100.
      OXSCFM = OXAIR*379./(60.*29.)
      SRCAO = ASHREM*(ASHCAO/100.)/(ACID*MCAO)
      SRMGA = ASHREM*(ASHMGO/100.)/(ACID*MMGO)
      FMGADD = (WPMGO+MGOADD)*MCACO3/(WPLS*MMGO)
      IF (ISR.EQ.1) GO TO 1000
      SRADD = SR
      IF (SRADD.LT.0.) SRADD = 0.
      SRMGO = SRMGA + SRADD*FMGADD
      SRTOT = SRCAO + SRMGO + SRADD
      GO TO 1010
 1000 CONTINUE
      SRTOT = SR
      IF (FMGADD.LT.EPS) SRMGO = 0.
      IF (FMGADD.GE.EPS) SRMGO = SRMGA + (SRTOT-SRCAO)/(1.+(1./FMGADD))
      SRADD = SRTOT-SRCAO-SRMGO
      IF (SRADD.LT.0.) SRADD = 0.
 1010 CONTINUE
      WRITE(8,*)' SRTOT,SRCAO,SRMGO,ACID,MCA',SRTOT,SRCAO,SRMGO,ACID,MCA
      LSCA = SRADD*ACID*MCA
      ASHCA = ASHREM*ASHCAO*MCA/(100.*MCAO)
      CATOT = LSCA + ASHCA
      LSADD = LSCA/((MCA/MCACO3)*(1.-FI-FMG))
      LIMEAD = LSADD*MCAO/MCACO3
      MGADD = MGOADD*MMG/MMGO
      FFMG = MMG/(100.*MMGO)
      WMG = LSADD*FFMG*(WPMGO+MGOADD) + ASHREM*ASHMGO*FFMG
      SO3 = SO2*(1.-FOX)*(MSO3/MSO2)
      SO4 = SO2*FOX*(MSO4/MSO2)
      CLX = (LBMH(2)*0.95/2. - WMG/MMG) * TREAT
      CLX = AMAX1(CLX,0.)
      CO3 = CATOT*MCO3/MCA - MCO3*(SO3/MSO3 + SO4/MSO4 + CLX)
      IF (IALK.EQ.0) INERTS = 0.
```

```
      IF (IALK.EQ.1) INERTS = LSADD*FI
      IF (IALK.EQ.2) INERTS = LIMEAD*FI
      INSOLS = INERTS + ASHREM*ASHFI
C     TOL = 0.03
      SO4L = MSO4*(WMG/MMG - 0.5*WCL/MCL)
      SO4L = AMAX1(SO4L,0.)
C     SO4CHK = TOL*SO4
C     IF (SO4L.LT.SO4CHK) SO4L = 0.
      SOLID = SO3*MCASO3/MSO3 + (SO4-SO4L)*MCASO4/MSO4 + CO3*MCACO3/MCO3
     1 + INSOLS
      LIQUID = SOLID*(100.-PSD)/PSD
      H2O = LIQUID - WMG - WCL - SO4L
      FH2O = 55.5/(H2O/MH2O)
      MGCONC = (WMG/MMG)*FH2O*TREAT
      CLCONC = (WCL/MCL)*FH2O*0.95*TREAT
      PPCO2 = 0.0003
      JPRINT = JEQ1
      TC = (5./9.)*(TWG-32.)
C
      CALL EQCALL (JPRINT,MGCONC,CLCONC,PPCO2,TC,X,PHP)
C
      PHPOND = PHP
      CAPPM = 1000.*MCA*(X(11)+X(12)+X(13)+X(14)+X(15)+X(16))
      MGPPM = 1000.*MMG*MGCONC
      SO3PPM = 1000.*MSO3*(X(6)+X(7)+X(8)+X(15)+X(21))
      SO4PPM = 1000.*MSO4*(X(9)+X(10)+X(16)+X(22))
C     CLPPM = 1000.*MCL*CLCONC
      CLPPM = (WCL/LIQUID)*1.E6*0.95*TREAT
      CH2O = H2O/1.E6
      SO3S = SO3 - CH2O*SO3PPM
      SO4S = SO4 - CH2O*SO4PPM
      SO3S = AMAX1(SO3S,0.)
      SO4S = AMAX1(SO4S,0.)
      CASO3 = SO3S*MCASO3/MSO3
      CASO4 = SO4S*MCASO4/MSO4
      FLIQ = H2O/(MH2O*55.5*1000.)
      CALIQ = FLIQ*CAPPM
      CACO3 = (CATOT - CASO3*MCA/MCASO3 - CASO4*MCA/MCASO4 - CALIQ)
     1 *MCACO3/MCA
      CACO3 = AMAX1 (CACO3,0.)
      TDS = CAPPM + MGPPM + SO3PPM + SO4PPM + CLPPM + AD
      WFDS = TDS/1.E6
      H2O = (1.-WFDS)*(100.-PSD)*(CASO3+CASO4+CACO3+INSOLS)/PSD
      WDIS = CASO3 + CASO4 + CACO3 + INSOLS + H2O*(1./(1.-WFDS))
      CLPPM = WCL*0.95*TREAT*1.E6/WDIS/(1.-PSD/100.)
      DGPM = WDIS*(1.-0.00615*PSD)/(60.*8.33)
      RETURN
      END
```

```
$NOFLOATCALLS
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C      SUBPROGRAM SRMOD
C      REVISION DATE 82-01-25
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       SUBROUTINE SRMOD(ICR)
C
       COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
      1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
      2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
      3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
      4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
      5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
      6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
       COMMON/COMPLE/ICOMPL, IRPT, IRUN, NEXT, IEQPR, IWTBAL, ITM, INPOPT, NOPART
       COMMON/EMIT/ISO2, XSO2, ISSO2, ELISO2, SO2REF, SO2LMB, SO2PPM
       COMMON/HGAS/ACFM, THG, YCO2, YHCL, YSO2, YO2, YN2, YH2O, ASH, WCL, WASH, IESP
       COMMON/LIQDIS/H2O, CAPPM, MGPPM, SO3PPM, SO4PPM, CLPPM, TDS, PHPOND, DGPM
       REAL MGPPM
       COMMON/LS/IALK, ISR, SR, SRADD, WPMGO, WPI, WPM, LSADD, LIMEAD, MGOADD, WMG
       REAL LSADD. LIMEAD, MGOADD
       COMMON/OUTPT/JOUTPT, JIT, JEQ1, JEQ2, JEQ3
       COMMON/RTIME/TR, ALCONT(9)
       COMMON/SCRUB/ISCRUB, IFOX, IADD, VLG, VTR, IFAN
       COMMON/SCRUBR/LG, V, PPMOUT, SO2R, ELIMIT, PR, PD, WPS, OX, PH, NSTAGE,
      1       NGRID, HS, ACS, ACSTRN
       REAL LG
C
       REAL NTANKS, MGCLQ, L
       REAL NTUST, NTUV
       DATA C1, C2, C3, C4, C5, C6/1.2, 1.5, 0.75, -8.8E-5, -0.12, 1./
       DATA D1, D2, D3/6.1E-5, 0.65, 1.5/
       DATA DS/1.5/, NTANKS/1./
       DATA V1, V2, V3/3.66E-3, 0.60, 1.20/
       DATA W1, W2, W3, W4/1.5E-3, 1., 0.40, 0.19/
C
C      STOICHIMETREY, OR PH, OR L/G MODEL
C        BECHTEL CONSTANTS 2-06-78,  PH AND SR
C
       ITMAX = 50
       ITPH = 0
       ITX = 0
       XNS = FLOAT(NSTAGE)
       XNGRID = FLOAT(NGRID)
       FHS = (XNS*HS/DS)+XNGRID
       MGCLQ = MGPPM-CLPPM/2.92
       MGCLQ = AMAX1(MGCLQ, 0.)
       L=LG*V*60./1000.
```

```
      IF(ICR.EQ.2) GO TO 1220
      IF (ISR.EQ.3.OR.ISCRUB.GE.3) GO TO 1000
      F1 = ALOG(100./(100.-SO2R))
      F2 = ALOG(F1)
 1000 CONTINUE
      IF (IALK.EQ.2) GO TO 1050
      IF (ISCRUB.EQ.1.OR.ISCRUB.EQ.3.OR.ISCRUB.EQ.4) GO TO 1010
C     TCA LIMESTONE MODEL CONSTANTS
C     (SEE P 14-70 OF THIRD PROGRESS REPORT FOR ADVANCED PROGRAM)
C     ALL LIMESTONE CASES WILL USE THIS MODEL    ----    SDG
C     NEW SCRUBBING CONSTANTS C.ROWLAND 3-6-80
      A1 = 2.06E-3
      A2 = 0.72
      A3 = 0.0
      A4 = 0.045
      A5 = 0.72
      A6 = 8.4E-5
      A7 = -1.7E-4
      A8 = 0.0
      IF (ISCRUB.GE.3) GO TO 1020
      GO TO 1130
 1010 CONTINUE
      A1=1.37E-3
      A2=1.
      A3=-1.
      A4=0.
      A5=1.
      A6=1.25E-4
      A7=0.
      A8=0.
      IF (ISCRUB.LT.3) GO TO 1040
 1020 CONTINUE
      NTUV = V1*(VLG**V2)*(VPD**V3)
      IF (ISR.EQ.3) GO TO 1170
      SO2RV = 1.-EXP(-NTUV)
      SO2RT = SO2R/100.
      SO2RST = (SO2RT-SO2RV)/(1.-SO2RV)
      IF (SO2RST.GT.0.) GO TO 1030
      GO TO 1080
 1030 CONTINUE
      F1 = ALOG(1./(1.-SO2RST))
      F2 = ALOG(F1)
 1040 CONTINUE
      IF (ISR.EQ.2) GO TO 1200
      GO TO 1130
 1050 CONTINUE
C     TCA LIME MODEL CONSTANTS
C     (SEE P 14-71 OF ABOVE)
C     ALSO VENTURI SPRAY TOWER WITH LIME    ---    SDG
      IF (ISCRUB.EQ.1.OR.ISCRUB.EQ.3.OR.ISCRUB.EQ.4) GO TO 1060
      A1 = 0.0187
      A2 = 1.
```

```
          A3 = 0.
          A4 = 0.039
          A5 = 0.16
          A6 = 1.6E-4
          A7 = -2.9E-4
          A8 = 0.
          IF (ISCRUB.GE.3) GO TO 1070
          IF (ISR.EQ.2) GO TO 1210
          GO TO 1130
   1060 CONTINUE
C         SPRAY TOWER ONLY - LIME MODEL CONSTANTS
C         A1 = 0.082
          A1 = 0.116
          A2 = 1.0
          A3 = -1.36
          A4 = 0.
          A5 = 0.29
          A6 = 3.2E-4
          A7 = -9.5E-5
C         A8 = 5.2E-5
          A8 = 0.0
          IF (ISCRUB.LT.3) GO TO 1120
   1070 CONTINUE
          PHOLD = PH
          IF (IALK.EQ.1) NTUV = V1*(VLG**V2)*(VPD**V3)
          IF (IALK.EQ.2) NTUV = W1 * (VLG**W2)*(VPD**W3)*EXP(W4*PH)
          SO2RV = 1.-EXP(-NTUV)
          IF (ISR.EQ.3) GO TO 1180
          SO2RT = SO2R/100.
          SO2RST = (SO2RT-SO2RV)/(1.-SO2RV)
          IF (SO2RST.GT.0.) GO TO 1110
          ITX = ITX+1
          IF (ITX.GT.50) GO TO 1090
          PH = PH-0.2
          GO TO 1070
   1080 CONTINUE
          WRITE(6,2000)
   2000 FORMAT(/1H ,'****     SO2RST.LE.0     CASE ENDED'/)
          GO TO 1100
   1090 CONTINUE
          WRITE(6,2050)
   2050 FORMAT(/1H,'****  ITX.GT.50  CASE ENDED'/)
   1100 CONTINUE
          ICOMPL=1
          RETURN
C
   1110 CONTINUE
          F1 = ALOG(1./(1.-SO2RST))
          F2 = ALOG(F1)
   1120 CONTINUE
          IF (ISR.EQ.2) GO TO 1210
   1130 CONTINUE
```

```
      IF (ISR.EQ.3) GO TO 1170
      V=ABS(V)
      L=ABS(L)
      PHX = F2-ALOG(A1)-A2*ALOG(L)-A3*ALOG(V)-A4*FHS-A6*MGCLQ
     1 -A7*YSO2*1.E6
      PH = PHX/A5
      IF (ISCRUB.LT.3.OR.ISR.NE.1) GO TO 1160
      ITPH = ITPH+1
      IF (ITPH.GT.ITMAX) GO TO 1100
 2100 FORMAT (/1H ,'PH CALCULATION UNCONVERGED AFTER',I4,' ITERATIONS'/)
      IF (JIT.NE.1.OR.(IALK.EQ.1.AND.ISR.EQ.1))  GO TO 1140
      IF (ITPH.EQ.1) WRITE (6,2150)
 2150 FORMAT (/1H ,11X,'IT',6X,'PH',8X,'PHOLD')
      WRITE (6,2200) ITPH,PH,PHOLD
 2200 FORMAT (1H ,9X,I4,5X,F6.3,4X,F6.3)
 1140 CONTINUE
      IF(ISR.EQ.2) GO TO 1020
 1150 CONTINUE
      PHTOL = 0.005
      IF (ABS(PH-PHOLD).GT.PHTOL) GO TO 1070
 1160 CONTINUE
      IF (ISR.EQ.2) GO TO 1190
      RETURN
C
 1170 CONTINUE
      IF (ISR.EQ.3.AND.ISCRUB.GE.3) GO TO 1180
      SO2R = 100.-100.*EXP(-A1*(L**A2)*(V**A3)*
     1 EXP(A4*FHS+A5*PH+A6*MGCLQ+A7*YSO2*1.E6))
      RETURN
 1180 CONTINUE
      NTUST = A1*(L**A2)*(V**A3)*EXP(A5*PH+A6*MGCLQ+A7*YSO2*1.E6
     1 +A8*CLPPM)
      SO2R = 100.-100.*EXP(-(NTUV+NTUST))
      SO2RST = 1.-EXP(-NTUST)
      RETURN
C
 1190 CONTINUE
      EXPO = C3*PH+C4*MGCLQ+C5*FHS+C6/NTANKS
      L = F1*((C1*EXP(EXPO)/(SR-1.))**(1./C2))
      LG = L*1000./(V*60.)
      RETURN
C
 1200 CONTINUE
      PHX=ALOG(SR-1.)-ALOG(D1)-D2*F2
      PH=PHX/D3
 1210 CONTINUE
      XL = F2-ALOG(A1)-A3*ALOG(V)-A4*FHS-A5*PH-A6*MGCLQ-A7*YSO2*1.E6
     1 -A8*CLPPM
      L = EXP(XL/A2)
      LG=L*1000./(V*60.)
      RETURN
C
```

```
C       THESE STATEMENTS REPLACE THE STOICH SUBROUTINE
C
 1220 CONTINUE
      F1 = ALOG (100./(100.-SO2R))
      IF (ISCRUB.GE.3.AND.ISR.NE.2) F1=ALOG(1./(1.-SO2RST))
      F1L = F1/L
      FPOW = F1L**C2
      IF (ISR.GT.1) GO TO 1240
      IF (ISCRUB.EQ.1.OR.ISCRUB.EQ.3.OR.ISCRUB.EQ.4) GO TO 1230
C     CALCULATE STOICHIOMETRY RATIO
C     TCA WITH LIMESTONE
      SR = 1.+C1*FPOW*EXP(C3*PH+C4*MGCLQ+C5*FHS+C6/NTANKS)
      RETURN
C
 1230 CONTINUE
      SR=1.+D1*(F1**D2)*EXP(D3*PH)
      RETURN
C
 1240 CONTINUE
      IF (ISCRUB.EQ.1.OR.ISCRUB.EQ.3.OR.ISCRUB.EQ.4) GO TO 1250
      PHX = ALOG(SR-1.)-ALOG(C1)-ALOG(FPOW)-C4*MGCLQ-C5*FHS-C6*NTANKS
      PH = PHX/C3
      RETURN
C
 1250 CONTINUE
      PHX=ALOG(SR-1.)-ALOG(D1)-D2*ALOG(F1)
      PH=PHX/D3
      RETURN
      END
```

```
$NOFLOATCALLS
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C      SUBPROGRAM EQCALL
C      REVISION DATE 79-11-05
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       SUBROUTINE EQCALL(JPRINT, MG, CL, PPCO2, TC, X, PH)
C
       REAL MG
       DIMENSION IDENT(20), CCM(10), CW(10)
       DIMENSION X(50), CM(10), PP(2), SS(3), XMW(10)
       COMMON/LIQ2/IW4, AD, ADDC, ADLOSS, AMGLOS
C
       DATA XMW/40.08, 24.31, 22.99, 39.10, 80.06, 96.06, 60.01, 35.45, 146., 0.0/
C
       DO 1000 I = 1, 9
       CM(I) = 0.
       CCM(I) = 0.
       CW(I) = 0.
 1000  CONTINUE
       KK1 = 2
       JJS = 1
       PPH = 0.
       FNA = 0.
       SS(1) = 1.
       SS(3) = 1.
       SS1 = SS(1)
       SS3 = SS(3)
       PP(1) =0.
       PP(2) = PPCO2
       PSO2 = PP(1)
       PCO2 = PP(2)
       CM(1) = 0.25
       CM(3) = MG + 0.2
       CM(5) = 0.6 + CL
       CM(6) = MG
       CM(8) = CL
       CM(9) = AD /(XMW(9) * 1000.)
       CCM(1) = CM(5)
       CCM(2) = CM(6)
       CCM(5) = CM(1)
       CCM(6) = CM(3)
       CCM(8) = CM(8)
       DO 1010 I = 1, 8
       CW(I) = 1000.*CCM(I)*XMW(I)
1010   CONTINUE
       TK = TC + 273.16
       CALL BEQ(JPRINT, 1, 2, TK, X, CM, PP, SS, 0, PH, FNA)
       RETURN
```

END

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM BEQ
C     REVISION DATE 79-11-08
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE BEQ (NOPR,JSS,IOPT,TK,X,CM,PP,SS,IPH,PH,FNA)
C*************************************************************************
C****** THIS ROUTINE DETERMINES AQUEOUS SOLUTION EQUILIBRIA FOR THE
C       SYSTEM SO2,CO2,SO3,N2O5,CA,MG,NA,CL,ADIPIC ACID.
C
C****** FOR JSS=1 AND SPECIFIED PPCO2, SOLIDS ARE ALSO DETERMINED.
C
C*************************************************************************
C
C                        COMPONENTS
C                        (1)   H+
C                        (2)   OH-
C                        (3)   H2CO3
C                        (4)   HCO3-
C                        (5)   CO3--
C                        (6)   H2SO3
C                        (7)   HSO3-
C                        (8)   SO3--
C                        (9)   HSO4-
C                        (10)  SO4--
C                        (11)  CA++
C                        (12)  CAOH+
C                        (13)  CAHCO3+
C                        (14)  CACO3
C                        (15)  CASO3
C                        (16)  CASO4
C                        (17)  MG++
C                        (18)  MGOH+
C                        (19)  MGHCO3+
C                        (20)  MGCO3
C                        (21)  MGSO3
C                        (22)  MGSO4
C                        (23)  NO3-
C                        (24)  CANO3+
C                        (25)  NA+
C                        (26)  NANO3
C                        (27)  NAOH
C                        (28)  NAHCO3
C                        (29)  NACO3-
C                        (30)  NASO4-
C                        (31)  CL-
C                        (32)  CACO3 (S)
C                        (33)  CASO3 (S)
C                        (34)  CASO4 (S)
```

```
C                    (35)  CA(OH)2 (S)
C                    (36)  MGCO3 (S)
C                    (37)  MG(OH)2 (S)
C                    (38)  H2AD
C                    (39)  HAD-
C                    (40)  AD--
C                    (41)  CAAD
C
C               EQUATIONS
C               (1)    (H+)(OH-) = K(1)
C               (2)    (H2CO3) = PPCO2/K(2)
C               (3)    (H+)(HCO3-) = (H2CO3)*K(3)
C               (4)    (H+)(CO3--) = (HCO3-)*K(4)
C               (5)    (H+)(HSO3-) = (H2SO3)*K(5)
C               (6)    (H+)(SO3--) = (HSO3-)*K(6)
C               (7)    (H+)(SO4--) = (HSO4-)*K(7)
C               (8)    (CA++)(OH-) = (CAOH+)*K(8)
C               (9)    (CA++)(HCO3-) = (CAHCO3+)*K(9)
C               (10)   (CA++)(CO3--) = (CACO3)*K(10)
C               (11)   (CA++)(SO3--) = (CASO3)*K(11)
C               (12)   (CA++)(SO4--) = (CASO4)*K(12)
C               (13)   (MG++)(OH-) = (MGOH+)*K(13)
C               (14)   (MG++)(HCO3-) = (MGHCO3+)*K(14)
C               (15)   (MG++)(CO3--) = (MGCO3)*K(15)
C               (16)   (MG++)(SO3--) = (MGSO3)*K(16)
C               (17)   (MG++)(SO4--) = (MGSO4)*K(17)
C               (18)   SO2 MATERIAL BALANCE
C               (19)   SO3 MATERIAL BALANCE
C               (20)   CA MATERIAL BALANCE
C               (21)   MG MATERIAL BALANCE
C               (22)   CHARGE BALANCE
C               (23)   (H2SO3) = PPSO2/K(23)
C               (24)   (CA++)(NO3-) = (CANO3+)*K(24)
C               (25)   (NA+)(NO3-) = (NANO3)*K(25)
C               (26)   (NA+)(OH-) = (NAOH)*K(26)
C               (27)   (NA+)(HCO3-) = (NAHCO3)*K(27)
C               (28)   (NA+)(CO3--) = (NACO3-)*K(28)
C               (29)   (NA+)(SO4--) = (NASO4-)*K(29)
C               (30)   CO2 MATERIAL BALANCE
C               (31)   NO2 MATERIAL BALANCE
C               (32)   NA MATERIAL BALANCE
C               (33)   (CL-) = CONSTANT
C               (34)   (CA++)(CO3--) L.E. K(34)
C               (35)   (CA++)(SO3--) L.E. K(35)
C               (36)   (CA++)(SO4--) L.E. K(36)
C               (37)   (CA++)(OH-)**2 L.E. K(37)
C               (38)   (MG++)(CO3--) L.E. K(38)
C               (39)   (MG++)(OH-)**2 L.E. K(39)
C               (40)   (H+)(HAD-- = (H2AD)*K(40)
C               (41)   (H+)(AD--) = (HAD-)*K(41)
C               (42)   (CA++)(AD--) = (CAAD)*K(42)
C               (43)   AD MATERIAL BALANCE
```

```
      DIMENSION X(50),CM(10),PP(2),IFLAG(10),SS(3)
      COMMON/BIG1/BAGDLP,BAGRAT,BAGCOST,BGLIFE,PSSO2X,EBYPAS,RS,PSC,
     1 CCSA,EFFPS,ESPDLP,RESIST,SCARAT,FANHP,FANLC,FANMC,DELTAP,FILTSA,
     2 PSF,FILRAT,H,TL,VD,VE,TGASID,SKLAND,DRAINM,DRAINL,CLAYSK,WPSO3,
     3 WPSO4,WPCO3,WPINS,ASHCLN,WINDEX,PDMAX,FLOOD,SRAIR,OXAIR,OXSCFM,
     4 PRES,SO2RT,SO2RV,SO2RST,SDFEE,VPD,VT,HVCLN,HPTONW,
     5 ISLUDG,IFIXS,JSR,IVPD,IIIT,LON,LGMAX,ISTAND,NSBR,
     6 KEPASS,KPASO2,ICLAR,JSSO,JDIS,JSTR,JGPM,IRH,NSBF
      COMMON/GAMMA/GAM(50)
      COMMON/TEMP/AC,BC,CPK(2),CK(50)
      REAL K(50),LOW
      DIMENSION A(32),B(32),C(32),D(32)
C
      DATA  A/4470.99,0.,3404.71,2902.39,-843.67,-633.84,475.14,
     1 -273.,-301.85,-475.48,-504.8,2572.1,-517.99,-235.08,-504.8,
     2 -432.5,-1057.86,-1245.,0.,0.,0.,-303.41,-241.,-1660.,0.,4944.0,
     3 530.49,-955.14,-480.85,0.,0.,0./
C
      DATA  B/0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,23.15,0.,0.,0.,0.,0.,
     1 0.,0.,0.,0.,0.,0.,0.,0.,0.,37.745,-12.9722,0.,0.,0.,0.,0./
C
      DATA  C/.01706,0.,.032786,.02379,0.,0.,.018222,0.,0.,0.,0.,
     1 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,.032331,0.,0.
     2 ,0.,0.,0./
C
      DATA  D/6.0875,0.,14.8435,6.4980,-4.7171,-9.3320,5.0435,-2.29,
     1 -2.272,-4.7954,-5.091,63.60,-4.3223,-1.747,-5.091,-4.3715,-5.7950
     2 ,-4.49,.40,.57,.25,-2.2852,-1.529,-13.88,-6.347,105.36,-25.7085,
     3 -7.871,-12.514,-4.420,-5.495,-2.602/
C
      EPS = .000001
C     IF (JS.EQ.JSOLD.AND.LON.EQ.LONOLD.AND.ABS(TK-TOLD).LT.EPS) RETURN
      TC = TK-273.16
      TC2 = TC*TC
      DC = 87.740-0.40008*TC+9.398E-4*TC2-1.410E-6*TC2*TC
      DCTK = DC*TK
      SQDCTK = SQRT(DCTK)
      AC = 1.8246E6*2.3026/(DCTK*SQDCTK)
      BC = 50.29/SQDCTK
      DM = 2.3025851
      TLOG = ALOG10(TK)
      DO 1000 I=1,32
      CK(I) = EXP(DM*(-A(I)/TK-B(I)*TLOG-C(I)*TK+D(I)))
 1000 CONTINUE
      CPK(1) = EXP(DM*(1370./TK-4.51))
      CPK(2) = EXP(DM*(1015./TK-4.87))
C
      JS = JSS
 1002 CONTINUE
      X(1) = 0.
      EPS = .00000001
      IF (JS.NE.JSS) PP(1) = PP1
```

```
      PP1 = PP(1)
      PP2 = PP(2)
      LON = 0
      SCASO3 = 1000.
      CM15 = CM(1)-CM(5)
      CM35 = CM(3)-CM(5)
      IF (IOPT.EQ.3.OR.IOPT.EQ.1) CM(1) = 0.
      IF (IOPT.EQ.2.OR.IOPT.EQ.3) CM(2) = 0.
      IF (CM(4).GT.EPS.OR.CM(7).GT.EPS.OR.CM(8).GT.EPS.OR.CM(9).GT.EPS)
     1 LON = 1
      IT = 30
      TOL = 0.01
      IF (JS.NE.0) TOL = 0.0001
      IF (IPH.EQ.1) GO TO 1010
      ACID = CM(1)+CM(3)+0.5*CM(4)+0.5*CM(8)-CM(5)-CM(6)-0.5*CM(7)
      IF (ACID.LE.0.) X(1) = 1.E-6
      IF (ACID.GT.0.) X(1) = 1.E-4
      IF (LON.EQ.1.AND.CM(7).GT.(CM(5)+CM(6)).AND.ACID.LT.0.) X(1) =
     1 1.E-8
      IF (PP(1).GT..0001) X(1) = 1.E-4
 1010 CONTINUE
      IF (JS.NE.JSS) GO TO 1030
      KOUNT = 0
      KN = 0
      KP = 0
      DO 1020 I=1,9
      IF(I.EQ.8) GO TO 1020
      IFLAG(I) = 0
      IF (CM(I).LT.EPS) GO TO 1020
      IFLAG(I) = 1
      KOUNT = KOUNT+1
      IF (I.GT.4.AND.I.LT.9) KP = KP+1
      IF (I.LT.5.OR.I.EQ.9) KN = KN+1
 1020 CONTINUE
      IF (KP.EQ.0.OR.KN.EQ.0) IT = 1
      IS5 = 1
      IF (JS.EQ.0.OR.IFLAG(5).EQ.0) IS5 = 0
C****** CALCULATE TEMPERATURE-DEPENDENT CONSTANTS
      EPS = .000001
C     IF (JS.EQ.JSOLD.AND.LON.EQ.LONOLD.AND.ABS(TK-TOLD).LT.EPS) GO TO 1030
      TC = TK-273.16
      TC2 = TC*TC
      DC = 87.740-0.40008*TC+9.398E-4*TC2-1.410E-6*TC2*TC
      DCTK = DC*TK
      SQDCTK = SQRT(DCTK)
      AC = 1.8246E6*2.3026/(DCTK*SQDCTK)
      BC = 50.29/SQDCTK
      DM = 2.3025851
      TLOG = ALOG10(TK)
      DO 1001 I=1,32
      CK(I) = EXP(DM*(-A(I)/TK-B(I)*TLOG-C(I)*TK+D(I)))
 1001 CONTINUE
```

```
      CPK(1) = EXP(DM*(1370./TK-4.51))
      CPK(2) = EXP(DM*(1015./TK-4.87))
C
C****** INITIAL ESTIMATES FOR CONCENTRATIONS TO DETERMINE
C       ACTIVITY COEFFICIENTS
 1030 CONTINUE
      DO 1040 I=2,42
      X(I) = 0.
 1040 CONTINUE
      X(7) = CM(1)
      X(10) = CM(3)
      X(11) = CM(5)
      X(17) = CM(6)
      X(23) = CM(4)
      X(25) = CM(7)
      X(31) = CM(8)
      X(39) = CM(9)
C****** OUTER LOOP FOR CONVERGENCE OF ACTIVITY COEFFICIENTS
      ITGAM = 2
      IF (JS.NE.0) ITGAM = 5
      DO 1450 JJ=1,ITGAM
      ITO = 0
      HI = 20.
      LOW = -1.
      CALL KCALC(JJ,JS,X,SS,K)
      IF (IPH.EQ.0) PH = -ALOG10(X(1)*GAM(1))
      IF (IPH.EQ.1) X(1) = 1./GAM(1)/(10.**PH)
      IF (IFLAG(2).EQ.0) X(3) = PP(2)/K(2)
      IF (IFLAG(1).EQ.0) X(6) = PP(1)/K(23)
C****** INNER LOOP FOR CONVERGENCE OF PH
 1050 CONTINUE
      X(2) = K(1)/X(1)
      IF (IFLAG(1).EQ.1) GO TO 1060
      X(7) = K(5)*X(6)/X(1)
      X(8) = K(6)*X(7)/X(1)
      GO TO 1070
 1060 CONTINUE
      X7X8 = X(1)/K(6)
      X6X8 = X7X8*X(1)/K(5)
      C1 = 1.+X7X8+X6X8
 1070 CONTINUE
      IF (IFLAG(2).EQ.1) GO TO 1080
      X(4) = K(3)*X(3)/X(1)
      X(5) = K(4)*X(4)/X(1)
      GO TO 1090
 1080 CONTINUE
      X4X5 = X(1)/K(4)
      X3X5 = X(1)*X(1)/K(4)/K(3)
      C2 = 1.+X4X5+X3X5
 1090 CONTINUE
      IF (IFLAG(3).EQ.0) GO TO 1100
      X9X10 = X(1)/K(7)
```

```
      C3 = 1.+X9X10
 1100 CONTINUE
      IF (IFLAG(5).EQ.0) GO TO 1120
      X12X11 = X(2)/K(8)
      IF (IFLAG(2).EQ.1) GO TO 1110
      X13X11 = X(4)/K(9)
      X14X11 = X(5)/K(10)
      C5 = 1.+X12X11+X13X11+X14X11
      GO TO 1120
 1110 CONTINUE
      C5 = 1.+X12X11
 1120 CONTINUE
      IF (IFLAG(6).EQ.0) GO TO 1140
      X18X17 = X(2)/K(13)
      IF (IFLAG(2).EQ.1) GO TO 1130
      X19X17 = X(4)/K(14)
      X20X17 = X(5)/K(15)
      C6 = 1.+X18X17+X19X17+X20X17
      GO TO 1140
 1130 CONTINUE
      C6 = 1.+X18X17
 1140 CONTINUE
      C7=0.0
      IF (IFLAG(7).EQ.0) GO TO 1160
      X27X25 = X(2)/K(26)
      IF (IFLAG(2).EQ.1) GO TO 1150
      X28X25 = X(4)/K(27)
      X29X25 = X(5)/K(28)
      C7 = 1.+X27X25+X28X25+X29X25
      GO TO 1160
 1150 CONTINUE
      C7 = 1.+X27X25
 1160 CONTINUE
      IF(IFLAG(9).EQ.0) GO TO 1170
      X39X40 = X(1)/K(41)
      X38X40 = X39X40*X(1)/K(40)
      C9 = 1. +X39X40 + X38X40
 1170 CONTINUE
      IF (JS.EQ.0) GO TO 1220
      IF (IFLAG(5).EQ.0.OR.JS.GE.2) GO TO 1200
      SCAOH2 = K(37)/X(2)/X(2)
      SCACO3 = K(34)/X(5)
      IF (PP(1).GT.1.E-20) SCASO3 = K(35)/X(8)
      IF (SCACO3.LT.SCASO3.AND.SCACO3.LT.SCAOH2) GO TO 1180
      IF (SCAOH2.LT.SCASO3) GO TO 1190
      SOL5 = SCASO3
      ISOL5 = 1
      GO TO 1200
 1180 CONTINUE
      SOL5 = SCACO3
      ISOL5 = 2
      GO TO 1200
```

```
 1190 CONTINUE
      SOL5 = SCAOH2
      ISOL5 = 3
 1200 CONTINUE
      IF (IFLAG(6).EQ.0) GO TO 1220
      SMGOH2 = K(39)/X(2)/X(2)
      SMGCO3 = K(38)/X(5)
      IF (SMGOH2.LT.SMGCO3) GO TO 1210
      SOL6 = SMGCO3
      ISOL6 = 2
      GO TO 1220
 1210 CONTINUE
      SOL6 = SMGOH2
      ISOL6 = 3
 1220 CONTINUE
      GO TO 1260
 1230 CONTINUE
      IF (JS.LT.2) GO TO 1260
      IF (JS.EQ.3) GO TO 1240
      IF (JS.EQ.4) GO TO 1250
C      CALL CASO3 (IFLAG,K,X,CM,PP,SOL6)
      IT = 5
      IF (IFLAG(3).EQ.0.AND.IFLAG(4).EQ.0.AND.IFLAG(6).EQ.0
     1 .AND.IFLAG(7).EQ.0) IT = 1
      CM15 = ABS(CM(1)-CM(5))
      CM15SQ = CM15*CM15
      IF (CM(5).GT.CM(1)) GO TO 11020
      X(11) = 0.
      X(17) = 0.
      X(25) = 0.
      DO 11010 I=1,IT
      IF (IFLAG(3).EQ.1) X(10) = CM(3)/(C3+X(11)/K(12)+X(17)/K(17)
     1 +X(25)/K(29))
      IF (IFLAG(4).EQ.1) X(23) = CM(4)/(1.+X(11)/K(24)+X(25)/K(25))
      F1 = C1+X(17)/K(16)
      F5 = C5+X(10)/K(12)+X(23)/K(24)
      DISC = CM15SQ+4.*K(35)*F5*F1
      X(8) = (CM15+SQRT(DISC))/(2.*F1)
      X(11) = K(35)/X(8)
      IF (IFLAG(6).EQ.0) GO TO 11000
      X(17) = CM(6)/(C6+X(8)/K(16)+X(10)/K(17))
      X(17) = AMIN1(X(17),SOL6)
11000 CONTINUE
      IF (IFLAG(7).EQ.1) X(25) = CM(7)/(C7+X(10)/K(29)+X(23)/K(25))
11010 CONTINUE
      F1 = C1+X(11)/K(11)+X(17)/K(16)
      RETURN
11020 CONTINUE
      X(8) = 0.
      X(10) = 0.
      X(23) = 0.
      DO 11040 I=1,IT
```

```
      IF (IFLAG(6).EQ.0) GO TO 11030
      X(17) = CM(6)/(C6+X(8)/K(16)+X(10)/K(17))
      X(17) = AMIN1(X(17),SOL6)
11030 CONTINUE
      IF (IFLAG(7).EQ.1) X(25) = CM(7)/(C7+X(10)/K(29)+X(23)/K(25))
      F1 = C1+X(17)/K(16)
      F5 = C5+X(10)/K(12)+X(23)/K(24)
      DISC = CM15SQ+4.*K(35)*F5*F1
      X(11) = (CM15+SQRT(DISC))/(2.*F5)
      X(8) = K(35)/X(11)
      IF (IFLAG(3).EQ.1) X(10) = CM(3)/(C3+X(11)/K(12)+X(17)/K(17)
     1 +X(25)/K(25))
11040 CONTINUE
      F1 = C1+X(11)/K(11)+X(17)/K(16)
      GO TO 1340
 1240 CONTINUE
c      CALL CASO4 (IFLAG,K,X,CM,PP,SOL6)
      IT = 5
      IF (IFLAG(1).EQ.0.AND.IFLAG(4).EQ.0.AND.IFLAG(6).EQ.0.AND.IFLAG(7)
     1.EQ.0) IT = 1
      CM35 = ABS(CM(3)-CM(5))
      CM35SQ = CM35*CM35
      IF (CM(5).GT.CM(3)) GO TO 21020
      X(11) = 0.
      X(17) = 0.
      X(25) = 0.
      DO 21010 I=1,IT
      IF (IFLAG(1).EQ.1) X(8) = CM(1)/(C1+X(11)/K(11)+X(17)/K(16))
      IF (IFLAG(4).EQ.1) X(23) = CM(4)/(1.+X(11)/K(24)+X(25)/K(25))
      F3 = C3+X(17)/K(17)+X(25)/K(29)
      F5 = C5+X(8)/K(11)+X(23)/K(24)
      DISC = CM35SQ+4.*K(36)*F5*F3
      X(10) = (CM35+SQRT(DISC))/(2.*F3)
      X(11) = K(36)/X(10)
      IF (IFLAG(6).EQ.0) GO TO 21000
      X(17) = CM(6)/(C6+X(8)/K(16)+X(10)/K(17))
      X(17) = AMIN1(X(17),SOL6)
21000 CONTINUE
      IF (IFLAG(7).EQ.1) X(25) = CM(7)/(C7+X(10)/K(29)+X(23)/K(25))
21010 CONTINUE
      F3 = C3+X(11)/K(12)+X(17)/K(17)+X(25)/K(29)
      RETURN
21020 CONTINUE
      X(8) = 0.
      X(10) = 0.
      X(23) = 0.
      DO 21040 I=1,IT
      IF (IFLAG(6).EQ.0) GO TO 21030
      X(17) = CM(6)/(C6+X(8)/K(16)+X(10)/K(17))
      X(17) = AMIN1(X(17),SOL6)
21030 CONTINUE
      IF (IFLAG(7).EQ.1) X(25) = CM(7)/(C7+X(10)/K(29)+X(23)/K(25))
```

```
      F3 = C3+X(17)/K(17)+X(25)/K(29)
      F5 = C5+X(8)/K(11)+X(23)/K(24)
      DISC = CM35SQ+4.*K(36)*F5*F3
      X(11) = (CM35+SQRT(DISC))/(2.*F5)
      X(10) = K(36)/X(11)
      IF (IFLAG(1).EQ.1) X(8) = CM(1)/(C1+X(11)/K(11)+X(17)/K(16))
21040 CONTINUE
      F3 = C3+X(11)/K(12)+X(17)/K(17)+X(25)/K(29)
      GO TO 1340
 1250 CONTINUE
C      CALL CASOX (IFLAG,K,X,CM,PP,SOL6)
      IT = 5
      IF (IFLAG(4).EQ.0.AND.IFLAG(6).EQ.0.AND.IFLAG(7).EQ.0) IT = 1
      CM135 = ABS(CM(1)+CM(3)-CM(5))
      C135SQ = CM135*CM135
      IF (CM(5).GT.(CM(1)+CM(3))) GO TO 31020
      X(11) = 0.
      X(17) = 0.
      X(25) = 0.
      DO 31010 I=1,IT
      IF (IFLAG(4).EQ.1) X(23) = CM(4)/(1.+X(11)/K(24)+X(25)/K(25))
      F1 = C1+X(17)/K(16)
      F3 = C3+X(17)/K(17)+X(25)/K(29)
      F5 = C5+X(23)/K(24)
      THF = F1+F3*K(36)/K(35)
      DISC = C135SQ+4.*K(35)*F5*THF
      X(8) = (CM135+SQRT(DISC))/(2.*THF)
      X(11) = K(35)/X(8)
      X(10) = K(36)/X(11)
      IF (IFLAG(6).EQ.0) GO TO 31000
      X(17) = CM(6)/(C6+X(8)/K(16)+X(10)/K(17))
      X(17) = AMIN1(X(17),SOL6)
31000 CONTINUE
      IF (IFLAG(7).EQ.1) X(25) = CM(7)/(C7+X(10)/K(29)+X(23)/K(25))
31010 CONTINUE
      F1 = C1+X(11)/K(11)+X(17)/K(16)
      F3 = C3+X(11)/K(12)+X(17)/K(17)+X(25)/K(29)
      RETURN
31020 CONTINUE
      X(8) = 0.
      X(10) = 0.
      X(23) = 0.
      DO 31040 I=1,IT
      IF (IFLAG(6).EQ.0) GO TO 31030
      X(17) = CM(6)/(C6+X(8)/K(16)+X(10)/K(17))
      X(17) = AMIN1(X(17),SOL6)
31030 CONTINUE
      IF (IFLAG(7).EQ.1) X(25) = CM(7)/(C7+X(10)/K(29)+X(23)/K(25))
      F1 = C1+X(17)/K(16)
      F3 = C3+X(17)/K(17)+X(25)/K(29)
      F5 = C5+X(23)/K(24)
      DISC = C135SQ+4.*F5*(F1*K(35)+F3*K(36))
```

```
      X(11) = (CM135+SQRT(DISC))/(2.*F5)
      X(8) = K(35)/X(11)
      X(10) = K(36)/X(11)
31040 CONTINUE
      F1 = C1+X(11)/K(11)+X(17)/K(16)
      F3 = C3+X(11)/K(12)+X(17)/K(17)+X(25)/K(29)
      GO TO 1340
 1260 CONTINUE
C****** CALCULATE BASE IONS (SO3--,CO3--,SO4--,NO3-,CA++,MG++,NA+)
      IF (IFLAG(1).EQ.1) X(8) = 0.
      IF (IFLAG(2).EQ.1) X(5) = 0.
      X(10) = 0.
      X(23) = 0.
      X(40) = 0.
      F51 = 0.
      F53 = 0.
      C5S = C5
      C6S = C6
      C7S = C7
      DO 1320 I=1,IT
      IF (IFLAG(2).EQ.0) GO TO 1270
      X514 = X(5)*X4X5
      IF (IFLAG(5).EQ.1) C5 = C5S+X(5)/K(10)+X514/K(9)
      IF (IFLAG(6).EQ.1) C6 = C6S+X(5)/K(15)+X514/K(14)
      IF (IFLAG(7).EQ.1) C7 = C7S+X(5)/K(28)+X514/K(27)
 1270 CONTINUE
      IF (IFLAG(5).EQ.0) GO TO 1280
      X(11) = CM(5)/(C5+X(8)/K(11)+X(10)/K(12)+X(23)/K(24)+X(40)+K(42)
     1 +F51+F53)
      IF (JS.NE.0) X(11) = AMIN1(X(11),SOL5)
 1280 CONTINUE
      IF (IFLAG(6).EQ.0) GO TO 1290
      X(17) = CM(6)/(C6+X(8)/K(16)+X(10)/K(17))
      IF (JS.NE.0) X(17) = AMIN1(X(17),SOL6)
 1290 CONTINUE
      IF (IFLAG(7).EQ.1) X(25) = CM(7)/(C7+X(10)/K(29)+X(23)/K(25))
      IF (IFLAG(1).EQ.0) GO TO 1300
      F1 = C1+X(11)/K(11)+X(17)/K(16)
      X(8) = CM(1)/F1
      IF (IS5.EQ.0) GO TO 1300
      X(8) = AMIN1(X(8),K(35)/X(11))
      X(8) = AMAX1(X(8),CM15/F1)
      F51 = X(8)*(CM(1)-X(8)*F1)/K(35)
 1300 CONTINUE
      IF (IFLAG(2).EQ.1) X(5) = CM(2)/(C2+X(11)/K(10)+X(11)*X4X5/K(9)
     1 +X(17)/K(15)+X(17)*X4X5/K(14)+X(25)/K(28)+X(25)*X4X5/K(27))
      IF (IFLAG(3).EQ.0) GO TO 1310
      F3 = C3+X(11)/K(12)+X(17)/K(17)+X(25)/K(29)
      X(10) = CM(3)/F3
      IF (IS5.EQ.0) GO TO 1310
      X(10) = AMIN1(X(10),K(36)/X(11))
      X(10) = AMAX1(X(10),CM35/F3)
```

```
      F53 = X(10)*(CM(3)-X(10)*F3)/K(36)
 1310 CONTINUE
      IF (IFLAG(4).EQ.1) X(23) = CM(4)/(1.+X(11)/K(24)+X(25)/K(25))
 1320 CONTINUE
 1330 CONTINUE
C****** CALCULATE ALL UNDETERMINED IONS
 1340 CONTINUE
      IF (IFLAG(1).EQ.0) GO TO 1350
      X(7) = X(8)*X7X8
 1350 CONTINUE
      IF (IFLAG(2).EQ.0) GO TO 1360
      X(4) = X(5)*X4X5
 1360 CONTINUE
      IF (IFLAG(3).EQ.0) GO TO 1370
      X(9) = X(10)*X9X10
 1370 CONTINUE
      IF (IFLAG(5).EQ.0) GO TO 1380
      X(12) = X(11)*X12X11
      X(24) = X(11)*X(23)/K(24)
      X(13) = X(11)*X(4)/K(9)
 1380 CONTINUE
      IF (IFLAG(6).EQ.0) GO TO 1390
      X(18) = X(17)*X18X17
      X(19) = X(17)*X(4)/K(14)
 1390 CONTINUE
      IF (IFLAG(7).EQ.0) GO TO 1400
      X(29) = X(25)*X(5)/K(28)
      X(30) = X(25)*X(10)/K(29)
 1400 CONTINUE
      IF(IFLAG(9).EQ.0) GO TO 1410
      X(39) = X(40)*X39X40
 1410 CONTINUE
      IF (IPH.EQ.1) GO TO 1440
C****** CALCULATE NEW VALUE OF (H+)
      PHIS = PHI
      PHI = 2.*(X(17)+X(11)-X(5)-X(8)-X(10)-X(40))+X(1)+X(12)+X(13)+
     1 X(18)+X(19)+X(24)+X(25)-X(2)-X(4)-X(7)-X(9)-X(23)-X(29)-X(30)-
     2 X(31)-X(39)
      ITO = ITO+1
      IF (JJ.GT.1.AND.ITO.EQ.1.AND.ABS(PHI).LT.2.*ABS(PHIS)) GO TO 1440
      IF (PHI.LT.0.) GO TO 1430
      LOW = PH
      IF (HI.LT.19.) GO TO 1420
      PH = PH+1.
      X(1) = X(1)/10.
      GO TO 1050
 1420 CONTINUE
      IF ((HI-LOW).LT.TOL) GO TO 1440
      PH = (HI+LOW)/2.
      X(1) = 1./GAM(1)/(10.**PH)
      GO TO 1050
 1430 CONTINUE
```

```
      HI = PH
      IF (LOW.GT.0.) GO TO 1420
      PH = PH-1.
      X(1) = X(1)*10.
      GO TO 1050
 1440 CONTINUE
 1450 CONTINUE
C****** ITERATIVE CALCULATION COMPLETE
C
C****** CALCULATE NEUTRAL MOLECULES
      IF (IFLAG(1).EQ.0) GO TO 1460
      X(6) =   X(8)*X6X8
      PP(1) = X(6)*K(23)
 1460 CONTINUE
      IF (IFLAG(2).EQ.0) GO TO 1470
      X(3) = X(5)*X3X5
      PP(2) = X(3)*K(2)
 1470 CONTINUE
      X(14) = X(11)*X(5)/K(10)
      X(15) = X(11)*X(8)/K(11)
      X(16) = X(11)*X(10)/K(12)
      X(20) = X(17)*X(5)/K(15)
      X(21) = X(17)*X(8)/K(16)
      X(22) = X(17)*X(10)/K(17)
      X(26) = X(25)*X(23)/K(25)
      X(27) = X(25)*X27X25
      X(28) = X(25)*X(4)/K(27)
      X(38) = X(40)*X38X40
      X(41) = X(11)*X(40)/K(42)
      IF (JS.EQ.0) GO TO 1530
C****** CALCULATE SOLIDS
      EPS1 = 1.E-6
      IF (IFLAG(5).EQ.0) GO TO 1500
      IF (JS.EQ.3) GO TO 1480
      IF (IFLAG(1).EQ.1) X(33) = CM(1)-X(8)*F1
      IF (JS.EQ.2) GO TO 1500
 1480 CONTINUE
      IF (IFLAG(3).EQ.1) X(34) = CM(3)-X(10)*F3
      IF (JS.NE.1) GO TO 1500
      F5 = C5+X(8)/K(11)+X(10)/K(12)+X(23)/K(24)
      IF (ISOL5.EQ.1) X(33) = CM(5)-X(34)-X(11)*F5
      IF (ISOL5.EQ.2) X(32) = CM(5)-X(32)-X(34)-X(11)*F5
      IF (ISOL5.EQ.3) X(35) = CM(5)-X(33)-X(34)-X(11)*F5
      IF (X(33).GT.EPS1.OR.X(35).GT.EPS1) GO TO 1500
      IF (X(33).GT.EPS1) GO TO 1490
      IF (X(34).LT.EPS1) GO TO 1500
      JS = 3
      GO TO 1002
 1490 CONTINUE
      IF (ISOL5.EQ.1) GO TO 1500
      IF (X(34).LT.EPS1) JS=2
      IF (X(34).GT.EPS1) JS=4
```

```
      GO TO 1002
 1500 CONTINUE
      IF (IFLAG(6).EQ.0) GO TO 1510
      F6 = C6+X(8)/K(16)+X(10)/K(17)
      IF (ISOL6.EQ.2) X(36) = CM(6)-X(17)*F6
      IF (ISOL6.EQ.3) X(37) = CM(6)-X(17)*F6
 1510 CONTINUE
      DO 1520 I=32,37
      IF (X(I).LT.EPS1) X(I) = 0.
 1520 CONTINUE
 1530 CONTINUE
      SS(1) = X(8) * X(11) / K(35)
      SS(3) = X(10) * X(11) / K(36)
C****** PRINT OUTPUT
      IP = NOPR
      RETURN
      END
```

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM KCALC
C     REVISION DATE 79-11-05
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE KCALC (JJ,JS,X,SS,K)
C****** THIS ROUTINE DETERMINES ACTIVITY COEFFICIENTS, GAM, AND
C       EQUILIBRIUM CONSTANTS, K, FOR THE SO2,CO2,SO3,N2O5,CA,MG,NA,CL
C       SYSTEM.
C*******************************************************************************
      COMMON/BIG1/BAGDLP,BAGRAT,BAGCOST,BGLIFE,PSSO2X,EBYPAS,RS,PSC,
     1 CCSA,EFFPS,ESPDLP,RESIST,SCARAT,FANHP,FANLC,FANMC,DELTAP,FILTSA,
     2 PSF,FILRAT,H,TL,VD,VE,TGASID,SKLAND,DRAINM,DRAINL,CLAYSK,WPSO3,
     3 WPSO4,WPCO3,WPINS,ASHCLN,WINDEX,PDMAX,FLOOD,SRAIR,OXAIR,OXSCFM,
     4 PRES,SO2RT,SO2RV,SO2RST,SDFEE,VPD,VT,HVCLN,HPTONW,
     5 ISLUDG,IFIXS,JSR,IVPD,IT,LON,LGMAX,ISTAND,NSBR,
     6 KEPASS,KPASO2,ICLAR,JSSO,JDIS,JSTR,JGPM,IRH,NSBF
      COMMON/GAMMA/GAM(50)
      COMMON/TEMP/AC,BC,CPK(2),CK(50)
C
      DIMENSION A(11),B(11),C(41),CC(11),U(11),SS(3)
      REAL K(50),X(50)
      DATA  A/6.,3.,3.,4.5,4.5,3.,4.5,3.,2.,5.,4./
      DATA  B/.4,.3,.3,0.,0.,0.,.1,.3,-.2,.1,0./
      DATA  C/1.,1.,0.,1.,4.,0.,1.,4.,1.,4.,4.,1.,1.,0.,0.,0.,
     1 4.,1.,1.,0.,0.,0.,1.,1.,1.,0.,0.,0.,1.,1.,1.,0.,0.,0.,0.,0.,0.,0.
     2 ,1.,2.,0./
      DATA  CC/1.,1.,0.,1.,4.,4.,4.,4.,1.,1.,1./
      DATA  U/0.,0.,.076,0.,0.,0.,0.,0.,0.,0.,0./
C
      GAM(12) = 1.
      IF (JS.NE.0.AND.JJ.EQ.1) GO TO 1020
      FI = 0.
      DO 1000 I=1,41
      FI = FI+X(I)*C(I)
 1000 CONTINUE
      FI = FI/2.
      SFI = SQRT(FI)
      IT2 = 11
      IF (LON.EQ.0) IT2 = 8
      DO 1010 I=1,IT2
      DNM = 1.+BC*A(I)*SFI
      ACF = AC*CC(I)
      UM = 2.3026*U(I)
      GAM(I) = EXP(ACF*(-SFI/DNM+B(I)*FI)+UM*FI)
 1010 CONTINUE
      GO TO 1030
 1020 CONTINUE
      GAM(1) = 0.8
```

```
      GAM(2) = 0.8
      GAM(3) = 1.
      GAM(4) = 0.8
      GAM(5) = 0.4
      GAM(6) = 0.4
      GAM(7) = 0.4
      GAM(8) = 0.4
      GAM(9) = 0.8
      GAM(10) = 0.8
      GAM(11) = 0.8
 1030 CONTINUE
      K(1) = CK(1)/(GAM(1)*GAM(2))
      K(2) = GAM(3)/CPK(2)
      G314 = GAM(3)/(GAM(1)*GAM(4))
      K(3) = CK(3)*G314
      G415 = GAM(4)/(GAM(1)*GAM(5))
      K(4) = CK(4)*G415
      K(5) = CK(5)*G314
      K(6) = CK(6)*G415
      K(7) = CK(7)*GAM(2)/(GAM(1)*GAM(6))
      K(8) = CK(8)/GAM(7)
      K(9) = CK(9)*GAM(2)/(GAM(7)*GAM(4))
      G375 = GAM(3)/(GAM(7)*GAM(5))
      K(10) = CK(10)*G375
      K(11) = CK(11)*G375
      G76 = GAM(7)*GAM(6)
      K(12) = CK(12)*GAM(3)/G76
      K(13) = CK(13)/GAM(8)
      K(14) = CK(14)*GAM(2)/(GAM(8)*GAM(4))
      G385 = GAM(3)/(GAM(8)*GAM(5))
      K(15) = CK(15)*G385
      K(16) = CK(16)*G385
      K(17) = CK(17)*GAM(3)/(GAM(8)*GAM(6))
      K(23) = GAM(3)/CPK(1)
      IF (LON.EQ.1) GO TO 1060
      DO 1040 I=24,29
      K(I) = 1.
 1040 CONTINUE
      DO 1050 I =40,42
      K(I) = 1.
 1050 CONTINUE
      GO TO 1070
 1060 CONTINUE
      K(24) = CK(18)*GAM(2)/(GAM(7)*GAM(9))
      K(25) = CK(19)*GAM(3)/(GAM(10)*GAM(9))
      K(26) = CK(20)*GAM(3)/(GAM(2)*GAM(10))
      K(27) = CK(21)*GAM(3)/(GAM(4)*GAM(10))
      K(28) = CK(22)*GAM(2)/(GAM(5)*GAM(10))
      K(29) = CK(23)*GAM(2)/(GAM(6)*GAM(10))
      K(40) = CK(30)*GAM(3)/(GAM(1)*GAM(2))
      K(41) = CK(31)*GAM(2)/(GAM(1)*GAM(6))
      K(42) = CK(32)*GAM(3)/G76
```

```
1070 CONTINUE
     G75 = GAM(7)*GAM(5)
     K(34) = CK(24)/G75
     K(35) = CK(25)*SS(1)/G75
     K(36) = CK(26)*SS(3)/G76
     G22 = 1./(GAM(2)*GAM(2))
     K(37) = CK(27)*G22/GAM(7)
     K(38) = CK(28)/(GAM(8)*GAM(5))
     K(39) = CK(29)*G22/GAM(8)
     RETURN
     END
```

```
C
      LOGICAL FUNCTION VALID(DUM)
C
C          THIS FUNCTION DETERMINES THE VALIDITY OF A PARTICULAR IAPCS
C     USING THE FOLLOWING CRITERIA:
C          1 - METHOD 4 MAY NOT BE USED WITH METHODS 7 OR 9
C          2 - METHODS 5 OR 6 MAY NOT PRECEED (BUT MAY FOLLOW) 7 OR 9
C          3 - METHODS MUST BE IN ASCENDING NUMERICAL ORDER(EXCEPT AS IN
C              3 ABOVE)
C          4 - METHODS MAY NOT BE REPEATED IN THE SAME SYSTEM
C
C     THE CHECK IS MADE USING A FINITE STATE AUTOMATON REPRESENTED BY
C     ARRAY V.
C
      INTEGER FMB,CT,BOTTOM,TYPCT,CLNLVL
      LOGICAL KO(2),TVA,EX
      COMMON/INPUT/BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,VROM,FCROM,HHVCLN,PSCLN,PACLN,CSTCLN,
     &PNACLN,ALKCLN,CLCLN,FEOCLN,PMCLN,VCLN,FCCLN,FMB,NR,IS,CT,TYPCT,
     &CLNLVL,BOTTOM,NM,IAPCS(9),KO,TVA
      INTEGER*2 V(10,11),S
C
      DATA V/ 2,11, 3, 4, 5, 6, 7, 8, 9, 0,
     &        0, 2, 3, 4, 5, 6, 7, 8, 9,-1,
     &        0, 0, 0, 4, 5, 6, 7, 8, 9,-1,
     &        0, 0, 0, 0, 5, 6, 0,10, 0,-1,
     &        0, 0, 0, 0, 0, 6, 0,10, 0,-1,
     &        0, 0, 0, 0, 0, 0, 0,10, 0,-1,
     &        0, 0, 0, 0, 5, 6, 0, 8, 9,-1,
     &        0, 0, 0, 0, 0, 0, 0, 0, 9,-1,
     &        0, 0, 0, 0, 5, 6, 0, 0, 0,-1,
     &        0, 0, 0, 0, 0, 0, 0, 0, 0,-1,
     &        0, 0, 3, 4, 5, 6, 7, 8, 9,-1/
C
      S=1
C
C     LOOP ON INPUT SYSTEM CHECKING FOR VALIDITY
C
      DO 10 I=1,9
      S=V(IAPCS(I),S)
      IF(S) 30,20,10
   10 CONTINUE
C
   20 WRITE(6,25)
   25 FORMAT(' ** INVALID IAPCS CONFIGURATION; SEE USER''S GUIDE. **')
      VALID=.FALSE.
      RETURN
   30 VALID=.TRUE.
      RETURN
      END
```

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C      SUBPROGRAM WETGAS
C      REVISION DATE 82-02-05
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       SUBROUTINE WETGAS
C
       COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
      1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
      2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
      3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
      4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
      5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
      6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
       COMMON/BYPAS2/BYLBH(6), BYLBMH(6), BSCFM, BACFM, PASSBY, XPASSD, IBY
       COMMON/HGAS/ACFM, THG, YCO2, YHCL, YSO2, YO2, YN2, YH2O, ASH, WCL, WASH, IESP
       COMMON/HGAS2/MPCT, LBMH, LBH, SCFM, HHG, HWG, CFR, BCFR, RCFR
       REAL MPCT(6), LBMH(6), LBH(6)
       COMMON/LS/IALK, ISR, SR, SRADD, WPMGO, WPI, WPM, LSADD, LIMEAD, MGOADD, WMG
       REAL LSADD, LIMEAD, MGOADD
       COMMON/SCRUBR/LG, V, PPMOUT, SO2R, ELIMIT, PR, PD, WPS, OX, PH, NSTAGE,
      1              NGRID, HS, ACS, ACSTRN
       REAL LG
       COMMON/SOLDIS/CASO3, CASO4, CACO3, INSOLS, WDIS, PSD, DENS, DEND
       REAL INSOLS
       COMMON/TRAN/NOTRAN, NOREDN, NTOTAL, PCNTRN, NORAW, NOUTIL, NOUNIT,
      1            NOPROD, NOSHIP, NSPREP, NSCOMP, NOCOMP, NOBYDU
       COMMON/WGAS/WACFM, WSCFM, TWG, ASHOUT, WASHOT, WMPCT, WLBMH, WLBH
       REAL WMPCT(6), WLBMH(6), WLBH(6)
       COMMON/WGCOMP/YWCO2, YWHCL, YWSO2, YWO2, YWN2, YWH2O, WVAPOR, NTRAIN
       REAL NTRAIN
C
       REAL MCO2, MHCL, MSO2, MO2, MN2, MH2O, MCACO3
C
       MCO2 = 44.01
       MHCL = 36.46
       MSO2 = 64.06
       MO2 = 32.
       MN2 = 28.02
       MH2O = 18.016
       MCACO3 = 100.1
       IF (IALK.EQ.2) CO2DIS = CACO3/MCACO3
       SO2ABS = YSO2*(SO2R/100.)*(SCFM*60./379.)
       HCLABS = YHCL*SCFM*60./379.
       IF (IALK.EQ.1) WLBMH(1) = LBMH(1)+SO2ABS+HCLABS
       IF (IALK.EQ.2) WLBMH(1) = LBMH(1)-CO2DIS
       WLBMH(2) = LBMH(2) * 0.05
       WLBMH(3) = LBMH(3)*(1.-(SO2R/100.))
       WLBMH(4) = LBMH(4)-SO2ABS*0.5*(OX/100.)
```

```
      WLBMH(5) = LBMH(5)
      WLBH(1)  = WLBMH(1)*MCO2
      WLBH(2)  = WLBMH(2)*MHCL
      WLBH(3)  = WLBMH(3)*MSO2
      WLBH(4)  = WLBMH(4)*MO2
      WLBH(5)  = WLBMH(5)*MN2
      WLBH(6)  = HWG*(WLBH(1)+WLBH(2)+WLBH(3)+WLBH(4)+WLBH(5))
      WLBMH(6) = WLBH(6)/MH2O
      WVAPOR = (WLBH(6)-LBH(6))/(60.*8.33)
      SUMMOL = WLBMH(1)+WLBMH(2)+WLBMH(3)+WLBMH(4)+WLBMH(5)+WLBMH(6)
      FM = 100./SUMMOL
      DO 1000 I=1,6
      WMPCT(I) = WLBMH(I)*FM
1000  CONTINUE
      YWCO2 = WMPCT(1)/100.
      YWHCL = WMPCT(2)/100.
      YWSO2 = WMPCT(3)/100.
      YWO2  = WMPCT(4)/100.
      YWN2  = WMPCT(5)/100.
      YWH2O = WMPCT(6)/100.
      WSCFM = SUMMOL*(379./60.)
      WACFM = WSCFM*(14.696/PRES)*(TWG+460.)/520.
      WASHOT = WASH*(1.-(PR/100.))
      IF(KPASO2.EQ.1) WASHOT = WASHOT*(1.-(PASSBY/100.))
      ASHOUT = WASHOT*7000./(60.*WSCFM)
      FACTNT=PCNTRN/(100.*8.34*60.)
      NTRAIN = FACTNT*(WLBH(1)+WLBH(2)+WLBH(3)+WLBH(4)+WLBH(5)+WLBH(6))
      RETURN
      END
```

```
$NOFLOATCALLS
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C      SUBPROGRAM STREAM
C      REVISION DATE 83-04-27
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       SUBROUTINE STREAM
C
       COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
      1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
      2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
      3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
      4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
      5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
      6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
       COMMON/FLOWS/GPM, MFLOW, WFLOW, TOTFLO
       REAL MFLOW(20,12), WFLOW(20,12), TOTFLO(20), GPM(20)
       COMMON/H2OAMT/SEEPRT, EVAPRT, RAIN, SEEP, EVAP, EVAPC, RAFAL, RAFALC,
      1              WLSM, H2OHYD, TWA, TWR
       COMMON/LIQDIS/H2O, CAPPM, MGPPM, SO3PPM, SO4PPM, CLPPM, TDS, PHPOND, DGPM
       REAL MGPPM
       COMMON/LIQ2/IW4, AD, ADDC, ADLOSS, AMGLOS
       COMMON/LS/IALK, ISR, SR, SRADD, WPMGO, WPI, WPM, LSADD, LIMEAD, MGOADD, WMG
       REAL LSADD, LIMEAD, MGOADD
       COMMON/OUTPT/JOUTPT, JIT, JEQ1, JEQ2, JEQ3
       COMMON/PND1/PDEPTH, PMXEXC, DISTPD, ILINER, CLAYIN, CLAYUC, HYPLMC,
      1            HYPLLC, PSAMAX, ACREC, APRAIN, SAPILE
       COMMON/SCRUB/ISCRUB, IFOX, IADD, VLG, VTR, IFAN
       COMMON/SCRUBR/LG, V, PPMOUT, SO2R, ELIMIT, PR, PD, WPS, OX, PH, NSTAGE,
      1              NGRID, HS, ACS, ACSTRN
       REAL LG, LSM
       COMMON/SOLDIS/CASO3, CASO4, CACO3, INSOLS, WDIS, PSD, DENS, DEND
       REAL INSOLS
       COMMON/WGAS/WACFM, WSCFM, TWG, ASHOUT, WASHOT, WMPCT, WLBMH, WLBH
       REAL WMPCT(6), WLBMH(6), WLBH(6)
C
       REAL M(14), LGCOOL, MACFM, MG
       DIMENSION WF(11), WFS(20), X(50), WR(11), WS(11), WT(11), XFLOW(20)
       DATA  M/129.1, 172.1, 100.1, 0., 18.016, 40.08, 24.31, 80.06, 96.06,
      1 35.45, 146., 40.31, 56.08, 74.1/
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C       STREAM NO.       DESCRIPTION
C            1             UNKNOWN - NOT USED
C            2             UNKNOWN - NOT USED
C            3             UNKNOWN - NOT USED
C            4             SYSTEM DISCHARGE
C            5             UNKNOWN
C            6             SCRUBBER SLURRY BLEED
```

```
C            7              TOTAL SUPERNATE RETURN
C            8              SUPERNATE OR MAKE-UP BALL MILL/SLAKER
C            9              LIME/LIMESTONE SLURRY FEED
C           10              SUPERNATE TO SCRUBBER OR EHT
C           11              RECYCLE SCRUBBER SLURRY
C           12              FLUE GAS COOLING SLURRY
C           13              THICKENER UNDERFLOW SLURRY
C           14              SUPERNATE OVERFLOW FROM THICKENER
C           15              SUPERNATE FROM POND ?
C           16              FILTER CAKE DISCHARGE
C           17              SUPERNATE RETURN FROM FILTER
C           18              UNKNOWN - NOT USED
C           19              UNKNOWN
C           20              NOT USED
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C         COMPONENT       DESCRIPTION
C            1             CASO3.1/2H2O SOLID
C            2             CASO4.2H2O   SOLID
C            3             CACO3        SOLID
C            4             INSOLUBLE
C            5             H2O
C            6             CA++
C            7             MG++
C            8             SO3--
C            9             SO4--
C           10             CL-
C           11             ADIPIC ACID AD--
C           12             NOT USED
C
C * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * * * * * * * *
C
      LSM = 0.0
      CONFAC = (1.-0.00615*WPS)/(8.33*60.)
      PSADD = 60.
      IF (IALK.EQ.2) PSADD = 20.
      LGCOOL = 4.
      IF (ISCRUB.GT.2) LGCOOL=VLG
      MACFM = WSCFM*(TWG+460.)*(14.696/PRES)/(520.*1000.)
      GPM(6) = WDIS*(PSD/WPS)*CONFAC
      EVAP=0.0
      EVAPC=0.0
      RAFAL=0.0
      RAFALC=0.0
      SEEP=0.0
      TCLARS=90.0
      TCLARD=90.0
      TPONDA=65.0
      DIFWAT = 0.0
      GPMDO = GPM(6)
      DO 1025 K = 1,5
      GPMD = GPM(6)
      IF (ISLUDG.EQ.1) GO TO 1000
```

```
      RAFALC=RAIN*CCSA*7.48/(12.0*8760.0*60.0)
      EVAPC=(1.0-0.75*WPS/100.0)*GPM(6)*(TWG-TCLARS)/1055.0
      IF (ISLUDG.GT.2) GO TO 1020
      EVAPCP=(1.0-0.75*PSC/100.0)*GPM(13)*(TCLARD-TPONDA)/1055.0
      GO TO 1010
 1000 CONTINUE
      EVAPCP=(1.0-0.75*WPS/100.0)*GPM(6)*(TWG-TPONDA)/1055.0
 1010 CONTINUE
      RAFAL=RAIN*APRAIN*7.48/(12.0*8760.0*60.0)
      SEEP=SEEPRT*60.0*7.48*APRAIN/(2.54*12.0)
      EVAP=EVAPRT*APRAIN*7.48/(12.0*8760.0*60.0)+EVAPCP
 1020 CONTINUE
      RECALW = DIFWAT
      DIFWAT = - RAFAL - RAFALC + EVAP + EVAPC
      DIFH20 = DIFWAT - RECALW
      GPM(6) = GPM(6) - DIFH20
 1025 CONTINUE
      RATIO = 1 - DIFWAT/GPM(6)
      RECYLE = RATIO
      RATIO = RATIO * (GPMDO/GPM(6))
      GPM(7)=GPM(6)-DGPM+RAFAL+RAFALC-EVAP-EVAPC-SEEP+DIFWAT
      XW = (100./WPS) -1.
      XC = (100./PSC) -1.
      XD = (100./PSD) -1.
      GROUP = (1.-XC/XW)/(1.-XD/XW)
      IF (ISLUDG-3) 1030,1040,1050
 1030 CONTINUE
      IW4 = 4
      GPM(4)=DGPM
      GPM(13)=0.
      GPM(14)=0.
      GPM(15)=GPM(7)
      GPM(16)=0.
      GPM(17)=0.
      GPM13 = 0.
      IF (ISLUDG.EQ.1) GO TO 1060
      GPM14 = (GPMDO-DGPM)*GROUP+RAFALC-EVAPC
      GPM(14) = (GPM(6)-DGPM)*GROUP+RAFALC-EVAPC+DIFWAT
      GPM13 = GPMDO-GPM14+RAFALC-EVAPC
      GPM(13) = GPM(6)-GPM(14)+RAFALC-EVAPC+DIFWAT
      GPM(15) = GPM(13)-DGPM+RAFAL-EVAP-SEEP
      IF (GPM(15).GE.0.0) GO TO 1060
      GPM(7)=GPM(7)-GPM(15)
      SEEP=SEEP+GPM(15)
      GPM(15)=0.0
      GO TO 1060
 1040 CONTINUE
      IW4 = 13
      GPM(4)=0.
      GPM(13)=DGPM
      GPM(14)=GPM(7)+DIFWAT*2.
      GPM(15)=0.
```

```
      GPM(16)=0.
      GPM(17)=0.
      GO TO 1060
 1050 CONTINUE
      IW4 = 16
      GPM(4)=0.
      GPM14=(GPMDO-DGPM)*GROUP+RAFALC-EVAPC
      GPM(14)=(GPM(6)-DGPM)*GROUP+RAFALC-EVAPC+DIFWAT
      GPM13=GPMDO-GPM14+RAFALC-EVAPC
      GPM(13)=GPM(6)-GPM(14)+RAFALC-EVAPC+DIFWAT
      GPM(15)=0.
      GPM(16)=DGPM
      GPM(17)=GPM(13)-DGPM
 1060 CONTINUE
      IF (GPM(7).GE.0.0) GO TO 1070
      SEEP=SEEP+GPM(7)
      GPM(7)=0.0
      GPM(15)=GPM(7)
 1070 CONTINUE
      IF (IALK.EQ.1) LSM = LSADD*(WPM/100.)/(8.33*60.)
      IF (IALK.EQ.2) LSM = LIMEAD*((WPM/100.)-(M(5)/M(13)))/(8.33*60.)
      IF (IALK.EQ.1) GPM(8) = LSADD*(100.-PSADD)/(PSADD*8.33*60.)
      IF (IALK.EQ.2) GPM(8) = LIMEAD*(M(14)/M(13))*(100.-PSADD)
     1 /(PSADD*8.33*60.)
      GPM(8) = GPM(8)-LSM
      IF (IALK.EQ.1) GPM(9) = (GPM(8)+LSM)*(1.-0.00615*PSADD)/(1.-(PSADD
     1 /100.))
      IF (IALK.EQ.2) GPM(9) = (GPM(8)+LSM)*(1.-0.00565*PSADD)/(1.-(PSADD
     1 /100.))
      GPM(10) = GPM(7)-GPM(8)
      IF (GPM(10).GE.0.0) GO TO 1080
      GPM(10)=-GPM(10)
      GPM(10)=0.0
 1080 CONTINUE
      GPM(11) = LG*MACFM
      GPM(12) = LGCOOL*MACFM
      GPM(19) = GPM(11)
      WFLS = 1.-(WPI/100.)-(WPMGO/100.)
      SWF = WPS/100.
      SOLID = CASO3+CASO4+CACO3+INSOLS
      FS = 100./SOLID
      WPSO3 = CASO3*FS
      WPSO4 = CASO4*FS
      WPCO3 = CACO3*FS
      WPINS = INSOLS*FS
      WF(1) = WPSO3/100.
      WF(2) = WPSO4/100.
      WF(3) = WPCO3/100.
      WF(4) = WPINS/100.
      WF(5) = 1.-(TDS/1.E6)
      WF(6) = CAPPM/1.E6
      WF(7) = MGPPM/1.E6
```

```
WF(8) = SO3PPM/1.E6
WF(9) = SO4PPM/1.E6
WF(10) = CLPPM/1.E6
WF(11) = AD/1.E6
WR(1) = WPSO3/100.
WR(2) = WPSO4/100.
WR(3) = WPCO3/100.
WR(4) = WPINS/100.
WR(5) = 1.-(TDS/1.E6) * RATIO
WR(6) = CAPPM/1.E6 * RATIO
WR(7) = MGPPM/1.E6 * RATIO
WR(8) = SO3PPM/1.E6 * RATIO
WR(9) = SO4PPM/1.E6 * RATIO
WR(10) = CLPPM/1.E6 * RATIO
WR(11) = AD/1.E6 * RATIO
WS(1) = WPSO3/100.
WS(2) = WPSO4/100.
WS(3) = WPCO3/100.
WS(4) = WPINS/100.
WS(5) = 1.-(TDS/1.E6)
WS(6) = CAPPM/1.E6 * RECYLE
WS(7) = MGPPM/1.E6 * RECYLE
WS(8) = SO3PPM/1.E6 * RECYLE
WS(9) = SO4PPM/1.E6 * RECYLE
WS(10) = CLPPM/1.E6 * RECYLE
WS(11) = AD/1.E6 * RECYLE
WFS(4) = PSD/100.
WFS(5) = 0.
WFS(6) = SWF
WFS(7) = 0.
WFS(8) = 0.
WFS(9) = PSADD/100.
WFS(10) = 0.
WFS(11) = SWF
WFS(12) = SWF
WFS(13) = PSC/100.
WFS(14) = 0.
WFS(15) = 0.
WFS(16)=PSF/100.0
WFS(17)=0.
IF (IALK.EQ.1) TOTFLO(9) = GPM(9)*8.33*60./(1.-WFS(9)*0.615)
IF (IALK.EQ.2) TOTFLO(9) = GPM(9)*8.33*60./(1.-WFS(9)*0.565)
DO 1100 I=4,17
IF (I.GT.12.AND.ISLUDG.EQ.1) GO TO 1100
IF (I.EQ.5.OR.(I.EQ.9.AND.IALK.EQ.2)) GO TO 1100
IF (I.NE.9) TOTFLO(I) = GPM(I)*8.33*60./(1.-WFS(I)*0.615)
DO 1090 I2=1,11
WT(I2) = WF(I2)
IF(I.EQ.11.OR.I.EQ.12) WT(I2) = WS(I2)
IF (I.EQ.9.AND.I2.LT.5) GO TO 1090
IF (I2.LT.5) PWF = WFS(I)
IF (I2.GT.4) PWF = 1.-WFS(I)
```

```
      WFLOW(I,I2) = TOTFLO(I)*PWF*WT(I2)
      IF (I.EQ.6)WFLOW(I,I2) = TOTFLO(I)*PWF*WR(I2)
      IF (I2.NE.4) MFLOW(I,I2) = WFLOW(I,I2)/M(I2)
 1090 CONTINUE
 1100 CONTINUE
      XFLOW(6) = GPMDO * 8.33 * 60./(1-WFS(6) * 0.615)
      XFLOW(13) = GPM13 * 8.33 * 60./(1-WFS(13) * 0.615)
      DO 1118 K = 6,13,7
      DO 1115 I2 = 1,5
      IF(I2.LT.5) PWF = WFS(K)
      IF(I2.GT.4) PWF = 1.-WFS(K)
      IF(I2.NE.5) WFLOW(K,I2) = XFLOW(K) * PWF * WR(I2)
      IF(I2.EQ.5.AND.K.EQ.6)WFLOW(K,5)=WFLOW(K,5)+(DIFWAT*8.33*60.)
      IF(I2.EQ.5.AND.K.EQ.13) WFLOW(K,5) = WFLOW(K,5)
      IF(I2.NE.4) MFLOW(K,I2) = WFLOW(K,I2)/M(I2)
 1115 CONTINUE
      TOTFLO(K) = 0.0
      DO 1116 I2 = 1,11
      TOTFLO(K) = TOTFLO(K) + WFLOW(K,I2)
 1116 CONTINUE
      IF(TOTFLO(K).LT.0.0) TOTFLO(K) = 0.0
      GPM(K) = TOTFLO(K) * (1-WFS(K)*0.615)/(8.33*60.)
 1118 CONTINUE
      TOTFLO(19) = TOTFLO(11)
      IF (IALK.EQ.2) GO TO 1130
      WFLOW(9,3) = TOTFLO(9)*WFLS*WFS(9)
      WFLOW(9,4) = TOTFLO(9)*(WPI/100.)*WFS(9)
      DO 1110 IS = 6,11
      WFLOW(9,IS) = WFLOW(8,IS)
      MFLOW(9,IS) = MFLOW(8,IS)
 1110 CONTINUE
      WFLOW(9,12) = TOTFLO(9)*(WPMGO/100.)*WFS(9)
      MFLOW(9,3) = WFLOW(9,3)/M(3)
      MFLOW(9,12) = WFLOW(9,12)/M(12)
      IF(IADD.NE.2) GO TO 1120
      ADLOSS = WFLOW(IW4,11) * ADDC
      WFLOW(9,11) = WFLOW(9,11) + ADLOSS
      MFLOW(9,11) = WFLOW(9,11) / M(11)
 1120 CONTINUE
      RETURN
 1130 CONTINUE
      JPRINT = JEQ3
      PPCO2 = 1.E-20
      TC = 80.
      MG = MGPPM*(GPM(8)/GPM(9))/(M(7)*1000.)
      CL = CLPPM*(GPM(8)/GPM(9))/(M(10)*1000.)
      CALL EQCALL (JPRINT,MG,CL,PPCO2,TC,X,PHADD)
      WF(6) = M(6)*(X(11)+X(12)+X(15)+X(16))/1000.
      WF(7) = M(7)*(X(17)+X(18)+X(21)+X(22))/1000.
      WF(8) = M(8)*(X(6)+X(7)+X(8)+X(15)+X(21))/1000.
      WF(9) = M(9)*(X(9)+X(10)+X(16)+X(22)+X(30))/1000.
      WF(5) = 1. - (WF(6)+WF(7)+WF(8)+WF(9)+WF(11))
```

```
      WF(10) = WF(10)*GPM(8)/GPM(9)
      PWF = 1. - WFS(9)
      WF(11) = WF(11)*GPM(8)/GPM(9)
      DO 1140 I2=5,11
      WFLOW(9,I2) = TOTFLO(9)*PWF*WF(I2)
      MFLOW(9,I2) = WFLOW(9,I2)/M(I2)
 1140 CONTINUE
      MFLOW(9,8) = AMIN1(MFLOW(9,8),MFLOW(8,8))
      WFLOW(9,8) = MFLOW(9,8)*M(8)
      MFLOW(9,9) = AMIN1(MFLOW(9,9),MFLOW(8,9))
      WFLOW(9,9) = MFLOW(9,9)*M(9)
      WFLOW(9,10) = WFLOW(8,10)
      MFLOW(9,10) = MFLOW(8,10)
      WFLOW(9,11) = WFLOW(8,11)
      MFLOW(9,11) = MFLOW(8,11)
      MFLOW(9,1) = MFLOW(8,8)-MFLOW(9,8)
      MFLOW(9,2) = MFLOW(8,9)-MFLOW(9,9)
      MFLOW(9,3) = (1./M(13))*LIMEAD*(1.-WPMGO/100.-WPI/100.)
     1 +MFLOW(8,6)-MFLOW(9,6)-MFLOW(9,1)-MFLOW(9,2)
      MFLOW(9,12) = LIMEAD*WPMGO/(100.*M(12)) + MFLOW(8,7) - MFLOW(9,7)
      WFLOW(9,1) = MFLOW(9,1)*M(1)
      WFLOW(9,2) = MFLOW(9,2)*M(2)
      WFLOW(9,3) = MFLOW(9,3)*M(14)
      WFLOW(9,4) = LIMEAD*WPI/100.
      WFLOW(9,12) = MFLOW(9,12)*M(12)
      IF(IADD.NE.2) GO TO 1150
      ADLOSS = WFLOW(IW4,11) * ADDC
      WFLOW(9,11) = WFLOW(9,11) + ADLOSS
      MFLOW(9,11) = WFLOW(9,11) / M(11)
 1150 CONTINUE
      RETURN
      END
```

```
$DEBUG
$NOFLOATCALLS
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C      SUBPROGRAM TVAIN
C      REVISION DATE 82-11-02
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       SUBROUTINE TVAIN(DIREXC)
C
       COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
      1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
      2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
      3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
      4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
      5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
      6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
        COMMON/OUTPT/JOUTPT, JIT, JEQ1, JEQ2, JEQ3
        COMMON/COMPLE/ICOMPL, IRPT, IRUN, NEXT, IEQPR, IWTBAL, ITM, INPOPT, NOPART
        COMMON/EQCOST/NOP(100), PEL(100), PEM(100), TOTPEL(9), TOTPEM(9)
        COMMON/CSTNDX/ICEPYR, CEPNDX, INVYR, IREVYR, RATLAB, RATMAT, ARM, ARL
        COMMON/HGAS/ACFM, THG, YCO2, YHCL, YSO2, YO2, YN2, YH2O, ASH, WCL, WASH, IESP
        COMMON/IND/ENGIN, ARCTEC, FLDEXP, FEES, CONT, START, CONINT, FILMAT(9)
        COMMON/RTIME/TR, ALCONT(9)
        COMMON/SCRUB/ISCRUB, IFOX, IADD, VLG, VTR, IFAN
        COMMON/SCRUBC/CSCRUB(14), SCRUBL, SCRUBM
        COMMON/SCRUBR/LG, V, PPMOUT, SO2R, ELIMIT, PR, PD, WPS, OX, PH, NSTAGE,
      1               NGRID, HS, ACS, ACSTRN
        REAL LG
        COMMON/TRAN/NOTRAN, NOREDN, NTOTAL, PCNTRN, NORAW, NOUTIL, NOUNIT,
      1           NOPROD, NOSHIP, NSPREP, NSCOMP, NOCOMP, NOBYDU
        COMMON/DUM/VERGPM(5), VERHD(5), VTOTLC(5), VTOTMC(5), VERTHP(8),
      &    AGITHP(10), AGITLC(10), AGITMC(10), DIATNK(10), HTTNK(10),
      &    TNKLC(10), TNKMC(10), VOLTNK(10), GPMPMP(11), PMPHP(11),
      &    TPMPLC(11), TPMPMC(11), TPH3, HPVAC, THKHT, CMECLC, CMECMC, CON3LG,
      &    CON8LG, DIASIL, DISPHD, FILTAR, HPMILL, HPSLAK, HTELE2, HTSILS,
      &    RAKEHP, REHTLC, REHTMC, SOBLLC, SOBLMC, SOBM, TFILTL, TFILTM, THIKAR,
      &    THKDIA, TPHSLK, TOTTKL, TOTTKM, VOLSIL, SOBL, NOTNK(10), NOMOT(45),
      &    NOPMOT(45), NSPMOT(45), NGR18, NOFILT, NOPREP, NOSILO, NOTHIK,
      &    NSFILT
        COMMON/DUMM/PMPFDN(11), TNKFDN(10), SPGR(10), BLDLAB(9), BLDMAT(9),
      1 EQULAB(9), EQUMAT(9), FDNLAB(9), FDNMAT(9), STRLAB(9), STRMAT(9),
      2 FBLDSF, FILFDL, FILFDM, HPFDRV, HPFPMP, VSCTNK, VERFDN, ELE2L, ELE2M,
      3 CSLAKL, CSLAKM, GPMILL, PCONL1, PCONL2, PCONL4, PCONM1, PCONM2, PCONM4,
      4 PEQLC1, PEQLC2, PEQMC1, PEQMC2, PSTLL1, PSTLL2, PSTLL3, PSTLL4, PSTLM1,
      5 PSTLM2, PSTLM3, PSTLM4, TPSLAK, TSLKHP, CMILLL, CMILLM, GPMTRN, NSUMPR
        COMMON/LANDIS/FILAND, TWLF, DISFUL, ELECLF, WATERF, FILANL, OVLF
        COMMON/LS/IALK, ISR, SR, SRADD, WPMGO, WPI, WPM, LSADD, LIMEAD, MGOADD, WMG
        COMMON/PCOST/POND(9), PDACRE, TOPHRS, PNDCAP, PNDLAB(9), PNDMAT(9), ELAN
        COMMON/PND1/PDEPTH, PMXEXC, DISTPD, ILINER, CLAYIN, CLAYUC, HYPLMC,
```

```
     1           HYPLLC,PSAMAX,ACREC,APRAIN,SAPILE
      REAL LSADD,LIMEAD,MGOADD
      DIMENSION SDLAND(4),ARLAND(9)
      COMMON/FIXFDN/ALCONY,FDNICS,PUGFDN,NSILFX,NASHS
      COMMON/SCROXD/COMPHP,COMCFM,SPRDIA,COPRMC,COPRLC,SPARMC,SPARLC,FNO
C
C
      DIMENSION DUCLAB(9),DUCMAT(9),ELELAB(9),ELEMAT(9),ENSLAB(9),
     1          ENSMAT(9),PIPLAB(9),PIPMAT(9),TOTLAB(9),TOTMAT(9)
      DIMENSION ARCHEC(9),CONEXP(9),CONTIN(9),
     1          CONTRA(9),DIRECT(9),DIREXC(9),ENGDES(9),
     2          ENTCON(9),PCONEX(9),PCONTR(9),PENGDE(9),
     3          PENTCO(9),PIRIND(9),PNDFIX(9),PONCAP(9)
      DIMENSION PONDIR(9),PONIND(9),PONTIN(9),PSTRTU(9),
     5          ROYALS(9),SALTAX(9),SERV(9),
     6          STRTUP(9),SUBCAP(9),SUBFIX(9),TAXEXC(9),TAXPND(9),
     7          TOTINV(9),TOTPCP(9),TOTPRC(9),WORCAP(9)
C
C
      FANHP=0.0
      FANMC=0.0
      FANLC=0.0
      CLAYSK=0.0
      VERFDN=0.0
      DO 1 I=1,42
      NOPMOT(I)=0
      NSPMOT(I)=0
1     CONTINUE
      DO 10 I =1,9
      PONDIR(I)=0.0
      PONIND(I)=0.0
      PONTIN(I)=0.0
      PNDFIX(I)=0.0
      PONCAP(I)=0.0
      PSTRTU(I)=0.0
      BLDMAT(I) = 0.0
      BLDLAB(I) = 0.0
      DUCMAT(I) = 0.0
      DUCLAB(I) = 0.0
      ELEMAT(I) = 0.0
      ELELAB(I) = 0.0
      ENSMAT(I) = 0.0
      ENSLAB(I) = 0.0
      EQUMAT(I) = 0.0
      EQULAB(I) = 0.0
      FDNMAT(I) = 0.0
      FDNLAB(I) = 0.0
      PIPLAB(I) = 0.0
      PIPMAT(I) = 0.0
      STRMAT(I) = 0.0
      STRLAB(I) = 0.0
      TOTMAT(I) = 0.0
```

```
      TOTLAB(I) = 0.0
      ARLAND(I) = 0.0
      WORCAP(I) = 0.0
      TOTPEL(I) = 0.0
      TOTPEM(I) = 0.0
   10 CONTINUE
      ISUBSW=1
  990 FORMAT(' ****    TVAIN CALLING SUBROUTINE #',I3,' ****')
C
      ISBCAL=1
      CALL EQUIPT
C     1   (SPGR,BLDLAB,BLDMAT,EQULAB,EQUMAT,FBLDSF,FDNLAB,FDNMAT,FILFDL,
C     2    FILFDM,HPFDRV,HPFPMP,NSUMPR,PMPFDN,STRLAB,STRMAT,TNKFDN,VSCTNK,
C     3    VERFDN)
C
      IF(ICOMPL.NE.0) GO TO 1020
C
      ISBCAL=2
C     STRUCTURAL STEEL
C      0.62 / MATERIAL, 1.10/LB LABOR
C
      STLVEN=0
      STLMUC=1240
      STLLUC=2200
C     SCRUBBER AREA
      VENSTL=0
      STLMEC=.001554*ACFM**.7163
      IF(IESP.NE.1)STLMEC=0
      SRBSTL=(VSCTNK/7.48)**.1188
      VENSTL=0.0
      IF (ISCRUB.LT.3) GO TO 1001
C     STRUCTURE COSTS FOR THE VENTURI ARE ESTIMATED AS 50% OF PROJECTED
C     STRUCTURE COSTS FOR A TCA OR SPRAY TOWER DESIGNED FOR A GAS
C     VELOCITY OF 12.5 FPS
      VENSTL=0.5*SRBSTL
 1001 CONTINUE
      ICSTRN=8
      IF(ACSTRN.GT.659) ICSTRN=9
      STLSRB=CSCRUB(ICSTRN)*SRBSTL
      STLVEN=CSCRUB(ICSTRN+5)*VENSTL
      STRMAT(4)=((STLMEC+STLSRB+STLVEN)*STLMUC)*RATMAT
      STRLAB(4)=((STLMEC+STLSRB+STLVEN)*STLLUC)*RATLAB
C
      ISBCAL=3
C     CONCRETE FOUNDATIONS
C     SCRUBBER AREA
      FDNMEC=.008622*ACFM**.6325
      IF(IESP.NE.1)FDNMEC=0
      FDNFAN=.000148*ACFM
      FDNSRB=.54386*STLSRB**.9988
      FDNVEN=.54386*STLVEN**.9988
      FDNEHT=TNKFDN(2)*NTOTAL
```

```
      FDNVHT=TNKFDN(8)*NTOTAL
      FDNOHT=TNKFDN(10)*NTOTAL
      FDNOXD=TNKFDN(9)*NOTNK(9)
C     CONCRETE FDN PUMPS  $180 M, $510 LABOR  1983
C      CONCRETE LARGE EQUIPMENT  $180 M, 312 LABOR1983
      FDNSPM=PMPFDN(4)+PMPFDN(5)
      FDNSCR=(FDNMEC+FDNSRB+FDNEHT+FDNVHT+FDNVEN)
      FOXFDN = PMPFDN(6) + FNO
      FOXFDL = FDNOHT + FDNOXD
      FDNMAT(3)=FDNFAN*171.*RATMAT
      FDNMAT(4)=(FDNSPM+FDNSCR)*180.*RATMAT
      FDNMAT(5)=(FOXFDN+FOXFDL)*180.*RATMAT
      FDNMAT(6)=0.0
      FDNLAB(3)= FDNFAN*312.*RATLAB
      FDNLAB(4)=(FDNSPM*510.  + FDNSCR*312.)*RATLAB
      FDNLAB(5)=(FOXFDN*510.  + FOXFDL*312.)*RATLAB
      FDNLAB(6)=0.0
C     SOLIDS DISPOSAL AREA
      FDNWAS=70.
      FDNWAT=0.0
      DO 1003 I=3,7
      J=I+4
      FDNWAS=FDNWAS+FDNICS+TNKFDN(I)*NOTNK(I)
      FDNWAT=FDNWAT + PMPFDN(J)
 1003 CONTINUE
C
C     FILTER CAKE CONVEYOR FOUNDATIONS AND RETAINER WALL
      FDNFCL=0.0
      FDNFCM=0.0
      IF(ISLUDG.LT.4) GO TO 1013
      FDNFCL=7785.00+4062.00
      FDNFCM=3848.00+1218.00
 1013 CONTINUE
C
      FDNLAB(7) = (FDNWAS*312. + (FDNWAT+VERFDN) * 510.)+FDNFCL+CLAYSK
      FDNMAT(7) =((FDNWAS + FDNWAT + VERFDN) * 180.)+FDNFCM
      FDNMAT(7) = FDNMAT(7)*RATMAT
      FDNLAB(7) = FDNLAB(7)*RATLAB
C      CALL FOUNDT
C     1    (NOTNK,FDNLAB,FDNMAT,FILFDL,FILFDM,PMPFDN,STLSRB,STLVEN,TNKFDN,
C     2      VERFDN)
C
      ISBCAL=4
      CALL PIPES
     1      (DCDIA,PIPLAB,PIPMAT)
C
      ISBCAL=5
      CALL DUCWRK
     1      (DCDIA,EXPLG,DUCLAB,DUCMAT,DUCTWD,HORNDL,
     2      PLENLG,SCRBLG,STKDIA)
C
      ISBCAL=6
```

```
      CALL INSTRM
     1       (ENSLAB,ENSMAT)
C
      ISBCAL=7
C
      PI = 3.1415926
      ACRELD = ACREC/43560.
      VENLEG=SCRBLG-4.0
      VENTWD=VENLEG/7.0
      ADDLAN=0.0
      IF(IADD.NE.0) ADDLAN=1000.
      NTPREP=NOPREP+NSPREP
      IF(IALK.GT.1) GO TO 1012
C     LIMESTONE PREP. LAND
      ARLAND(1)=((275+NGR18*100.)*225.+ADDLAN)* ACRELD
      GO TO 1022
 1012 CONTINUE
C     LIME PREP. LAND
      ADDLAN = ADDLAN/43560.
      ARLAND(1)=(0.1*TPH3+0.79+ADDLAN+(40*80/43560.))*ACREC
 1022 CONTINUE
      ARLAND(2)=((NTPREP*50.)*50. + ADDLAN) * ACRELD
      HORZDL=DIATNK(2)+SCRBLG/2+VENLEG/2+DIATNK(8)+DCDIA+EXPLG+35.
      ARLAND(4)=((SCRBLG+VENLEG)*NTOTAL*(VENTWD+5.)+(PI*(DIATNK(2)/2)
     1**2 + PI*(DIATNK(8)/2)**2)*NTOTAL) * ACRELD
      ARLAND(5)=(500.*NOCOMP + PI*NTOTAL*((DIATNK(9)/2)**2 +
     1(DIATNK(10)/2)**2))* ACRELD
      PLEN = PLENLG + DIATNK(2)
      ARLAND(3)=((PLEN*(DUCTWD+HORZDL))*ACRELD)-ARLAND(4)-ARLAND(5)
      ARLAND(6)= 0.0
      SDLAND(1)=0.
      SDLAND(2)=0.
      SDLAND(3)=0.
      SDLAND(4)=SKLAND/43560.
      IF(ISLUDG.GT.2) GO TO 1042
      SDLAND(1)=20.*DISTPD/43560.+PDACRE
      IF(ISLUDG.EQ.1) GO TO 1052
 1042 CONTINUE
      SDLAND(2)=(20.*1000.+2.*THKDIA*2.*THKDIA*NOTHIK)/43560.
      IF(ISLUDG.GE.4) SDLAND(3)=2.*FBLDSF/43560.
 1052 CONTINUE
      IF(ISLUDG.EQ.5) SDLAND(3) = SDLAND(3) + FILAND
      ARLAND(7)=(SDLAND(1)+SDLAND(2)+SDLAND(3)+SDLAND(4))*ACREC
C
      ISBCAL=8
      CALL ELECTR
     1       (REVHP,ELELAB,ELEMAT)
C
      ISBCAL=9
      CALL TOTALS
     1       (SERV,REVHP,ARCHEC,ARLAND,CONEXP,
     2       CONTIN,CONTRA,DIRECT,DIREXC,DUCLAB,DUCMAT,
```

```
     3       ELELAB,ELEMAT,ENGDES,ENSLAB,ENSMAT,
     4       ENTCON,PIPLAB,PIPMAT,PONDIR,PONIND,ROYALS,SALTAX,
     6       STRTUP,SUBCAP,SUBFIX,TAXEXC,TAXPND,
     7       TOTINV,TOTPCP,TOTPRC,WORCAP)
      IF (JOUTPT.EQ.2) GO TO 1000
C
      IF(IRPT.EQ.0) GO TO 1010
 1000 CONTINUE
C
      ISBCAL=10
C
      GO TO 1020
 1010 CONTINUE
C
      ISBCAL=11
 1020 continue
      RETURN
      END
```

```
$DEBUG
$NOFLOATCALLS
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM TO ESTIMATE EQUIPMENT COSTS
C     REVISION DATE 83-05-05
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE EQUIPT
C
      COMMON/DUM/VERGPM(5),VERHD(5),VTOTLC(5),VTOTMC(5),VERTHP(8),
     1   AGITHP(10),AGITLC(10),AGITMC(10),DIATNK(10),HTTNK(10),
     2   TNKLC(10),TNKMC(10),VOLTNK(10),GPMPMP(11),PMPHP(11),
     3   TPMPLC(11),TPMPMC(11),TPH3,HPVAC,THKHT,CMECLC,CMECMC,CON3LG,
     4   CON8LG,DIASIL,DISPHD,FILTAR,HPMILL,HPSLAK,HTELE2,HTSILS,
     5   RAKEHP,REHTLC,REHTMC,SOBLLC,SOBLMC,SOBM,TFILTL,TFILTM,THIKAR,
     6   THKDIA,TPHSLK,TOTTKL,TOTTKM,VOLSIL,SOBL,NOTNK(10),NOMOT(45),
     7  NOPMOT(45),NSPMOT(45),NGR18,NOFILT,NOPREP,NOSILO,NOTHIK,NSFILT
      COMMON/DUMM/PMPFDN(11),TNKFDN(10),SPGR(10),BLDLAB(9),BLDMAT(9),
     1 EQULAB(9),EQUMAT(9),FDNLAB(9),FDNMAT(9),STRLAB(9),STRMAT(9),
     2 FBLDSF,FILFDL,FILFDM,HPFDRV,HPFPMP,VSCTNK,VERFDN,ELE2L,ELE2M,
     3 CSLAKL,CSLAKM,GPMILL,PCONL1,PCONL2,PCONL4,PCONM1,PCONM2,PCONM4,
     4 PEQLC1,PEQLC2,PEQMC1,PEQMC2,PSTLL1,PSTLL2,PSTLL3,PSTLL4,PSTLM1,
     5 PSTLM2,PSTLM3,PSTLM4,TPSLAK,TSLKHP,CMILLL,CMILLM,GPMTRN,NSUMPR
      COMMON/ADDPRE/ADDSIL,DIAADD,HTSADD,IANGLE,ASILMC,ASILLC,PCOVMC,
     1PCOVLC,PCOVHP,ABINMC,ABINLC,AWFBMC,AWFBLC,ADUSTM,ADUSTL
      COMMON/BIG1/BAGDLP,BAGRAT,BAGCOST,BGLIFE,PSSO2X,EBYPAS,RS,PSC,
     1 CCSA,EFFPS,ESPDLP,RESIST,SCARAT,FANHP,FANLC,FANMC,DELTAP,FILTSA,
     2 PSF,FILRAT,H,TL,VD,VE,TGASID,SKLAND,DRAINM,DRAINL,CLAYSK,WPSO3,
     3 WPSO4,WPCO3,WPINS,ASHCLN,WINDEX,PDMAX,FLOOD,SRAIR,OXAIR,OXSCFM,
     4 PRES,SO2RT,SO2RV,SO2RST,SDFEE,VPD,VT,HVCLN,HPTONW,
     5 ISLUDG,IFIXS,JSR,IVPD,IT,LON,LGMAX,ISTAND,NSBR,
     6 KEPASS,KPASO2,ICLAR,JSSO,JDIS,JSTR,JGPM,IRH,NSBF
      COMMON/COMPLE/ICOMPL,IRPT,IRUN,NEXT,IEQPR,IWTBAL,ITM,INPOPT,NOPART
      COMMON/ADDSUM/APREPM,APREPL,ADMFND,ADSMAT,ADSLAB,ASM2,ASL2
      COMMON/EQCOST/NOP(100),PEL(100),PEM(100),TOTPEL(9),TOTPEM(9)
      COMMON/FIXCOS/FIXADD,DS,TFIXEQ,TFIXLB,HPFIX(8)
      COMMON/HGAS/ACFM,THG,YCO2,YHCL,YSO2,YO2,YN2,YH2O,ASH,WCL,WASH,IESP
      COMMON/LS/IALK,ISR,SR,SRADD,WPMGO,WPI,WPM,LSADD,LIMEAD,MGOADD,WMG
      COMMON/SCRUB/ISCRUB,IFOX,IADD,VLG,VTR,IFAN
      COMMON/SCRUBC/CSCRUB(14),SCRUBL,SCRUBM
      COMMON/SCROXD/COMPHP,COMCFM,SPRDIA,COPRMC,COPRLC,SPARMC,SPARLC,FNO
      COMMON/SRHT/TSTEAM,HVS,VRH,WSTEAM,ARH,DPRH,URH,Q,NTT,NB,PL,OD,
     &ARINCL,ARCORT,NINCLT,NCORTT,NINCLB,NCORTB,TEMPS
      DIMENSION ARH(6),DPRH(6),URH(6),NTT(6),NB(6),OD(6),ARINCL(6),
     &ARCORT(6),NINCLT(6),NCORTT(6),NINCLB(6),NCORTB(6)
      COMMON/SCRUBR/LG,V,PPMOUT,SO2R,ELIMIT,PR,PD,WPS,OX,PH,NSTAGE,
     &       NGRID,HS,ACS,ACSTRN
      REAL LG,INCONC,INCOLC
      COMMON/CSTNDX/ICEPYR,CEPNDX,INVYR,IREVYR,RATLAB,RATMAT,ARM,ARL
```

```
      DIMENSION XSCRUB(12,2)
      REAL LSADD,LIMEAD,MGOADD
      COMMON/TRAN/NOTRAN,NOREDN,NTOTAL,PCNTRN,NORAW,NOUTIL,NOUNIT,
     1            NOPROD,NOSHIP,NSPREP,NSCOMP,NOCOMP,NOBYDU
      COMMON/FILCON/FCCTLC,FCCTMC,HPFCCM,LNHFCC,LNIFCC
      COMMON/IND/ENGIN,ARCTEC,FLDEXP,FEES,CONT,START,CONINT,FILMAT(9)
      COMMON/FLOWS/GPM,MFLOW,WFLOW,TOTFLO
      REAL MFLOW(20,12),WFLOW(20,12),TOTFLO(20),GPM(20)
      COMMON/PND1/PDEPTH,PMXEXC,DISTPD,ILINER,CLAYIN,CLAYUC,HYPLMC,
     1            HYPLLC,PSAMAX,ACREC,APRAIN,SAPILE
      COMMON/RTIME/TR,ALCONT(9)
C
      DIMENSION VPMPMC(5),VMOTMC(5)
      DIMENSION TNKTOR(10),TSQFT(10),TLSQFT(10),UNITTG(10),PCSOLD(10)
C
      DATA XSCRUB/7089.7,4006.9,96.92,214.958,161.219,36.01,
     &            62.239,2.4126,.019413,484.306,3.6628,4.4708,
     &            .64497,.70419,1.03545,.9972,.99723,1.09583,
     &            .6765,.3008,1.0438,1.04483,.81783,.64795/
C
      ISUBSW=1
  990 FORMAT(' ****     EQUIPT CALLING SUBROUTINE #',I3,' ****')
C
      DO 1010 I=1,100
      PEM(I)=0.0
      PEL(I)=0.0
      NOP(I)=0
 1010 CONTINUE
      PEQLC1 = 0.0
      PEQLC2 = 0.0
      PEQLC3 = 0.0
      PEQLC4 = 0.0
      PEQMC1 = 0.0
      PEQMC2 = 0.0
      PEQMC3 = 0.0
      PEQMC4 = 0.0
      PSTLL1 = 0.0
      PSTLL2 = 0.0
      PSTLL3 = 0.0
      PSTLL4 = 0.0
      PSTLM1 = 0.0
      PSTLM2 = 0.0
      PSTLM3 = 0.0
      PSTLM4 = 0.0
      ISBCAL=1
      IF(IALK.EQ.1) CALL LSPREP
     1      (PEQLC3,PEQLC4,PEQMC3,PEQMC4)
C
      ISBCAL=2
      IF(IALK.EQ.2) CALL LIMEPR
C
      ADDSIL=0.0
```

```
      DIAADD=0.0
      HTSADD=0.0
      ASILMC=0.0
      ASILLC=0.0
      PCOVMC=0.0
      PCOVLC=0.0
      PCOVHP=0.0
      ADUSTM=0.0
      ADUSTL=0.0
      ABINMC=0.0
      ABINLC=0.0
      AWFBMC=0.0
      AWFBLC=0.0
      APREPM=0.0
      APREPL=0.0
      ADMFND=0.0
      ADSMAT=0.0
      ADSLAB=0.0
      COMPHP=0.0
      COMCFM=0.0
      NSCOMP=0
      NOCOMP=0
      SPRDIA=0.0
      COPRMC=0.0
      COPRLC=0.0
      SPARMC=0.0
      SPARLC=0.0
      RAKEHP=0.0
      HPVAC=0.0
      HPFDRV=0.0
      FNO=0.0
      HPFPMP=0.0
      FILFDM=0.0
      FILFDL=0.0
      TFILTM=0.0
      TFILTL=0.0
      TOTTKM=0.0
      TOTTKL=0.0
      THKHT=0.0
      NOTHIK=0
      NOFILT=0
      NSFILT=0
C
      ISBCAL=3
      IF(IADD.GT.0) CALL ADAMGO
     1                    (BLDLAB,BLDMAT)
C
      ISBCAL=4
      IF(ISLUDG.LE.1) GO TO 501
C
      PI=3.1415926
      NOTHIK = CCSA/125664.0 + 1.
```

```
      THIKAR = CCSA/NOTHIK
      THKDIA = SQRT(4.*THIKAR/PI)
      THKHT = 0.03*THKDIA + 4.
C     THICKENER TANK COSTS
      TKTKSF = PI*THKDIA*THKHT
      IF (THKDIA.GT.200.)GO TO 1006
      THKTKM = TKTKSF*8.00
      THKTKL = TKTKSF*5.57
      GO TO 1015
 1006 THKTKM = TKTKSF*9.60
      THKTKL = TKTKSF*6.75
 1015 CONTINUE
      EPAINT = 40.195 * (TKTKSF + 2*THIKAR)**0.562165
C     THICKENER BASIN COSTS  176 L, 307 M CONCRETE
      TBASCY = THIKAR*1/27
      TBASM = TBASCY*176.
      TBASL = TBASCY*307.
      THKRAD = THKDIA/2
C     THICKENER TUNNEL+ROOM COSTS  13*11*7
C     CONCRETE ROOM   180 M, 312 L  1983
      THROOM =(((13*7*2)+(13*11*2)+(7*11))*1.0)/27.
      TROOMM = THROOM * 180.
      TROOML = THROOM * 312.
      TUNM = 3900. + 40. * THKRAD
      TUNL = 7700. + 6.93 * THKRAD
C     THICKENER MECHNISM (RAKE) COSTS
      RAKEM = 381.39*THKDIA**1.25383
      RAKEL = 0.33*RAKEM
      RAKEHP = (THKDIA/140)**2*7.5
C     THICKENER EXCAVATION COSTS
      THKECV=(THKRAD*((12*5*11.23)+(7*5*4.31))+(11*13*7*11.23))
      THKECV=(THKECV + TBASCY*11.23) / 27
C     TOTAL THICKENER COSTS
      THIKM= THKTKM+TBASM+TUNM+RAKEM+EPAINT+TROOMM
      THIKL=THKTKL+TBASL+TUNL+RAKEL+THKECV+EPAINT+TROOML
      TOTTKM = THIKM*NOTHIK
      TOTTKL = THIKL*NOTHIK
  501 CONTINUE
C
      ISBCAL=5
      IF(ISLUDG.LT.4) GO TO 502
C
C     ROTARY DRUM, 33% SUBMERGENCE, 3/8 INCH CAKE
      PI=3.1415926
      NOFILT=1
      FILTAR=50.0
      IF (FILTSA.LE.50.0) GO TO 1005
      IF (FILTSA.GE.100.0) NOFILT=2
      IF (FILTSA.GT.1800.0) NOFILT=FILTSA/900.0+1.
      FILTAR=FILTSA/NOFILT
 1005 CONTINUE
C     SPARE FILTER
```

```
      NSFILT=1
      NTFILT=NOFILT+NSFILT
      FILTM = 10550.40*FILTAR**0.451152
      FILTL = 7174.75*FILTAR**0.19481
      TFILTM = FILTM*NTFILT
      TFILTL = FILTL*NTFILT
C
C     FILTER CAKE CONVEYOR
      LNIFCC=100
      LNHFCC=25*NTFILT
      FCCLEN=LNIFCC+LNHFCC
      FCCTLC=648.538+14.4572*FCCLEN
      FCCTMC=7914.39+174.967*FCCLEN
C
      FILTCN = FILTAR/(PI*27.)*NTFILT
      FILFDM = FILTCN * 180.
C     CONCRETE 180 M , 510 L  1983
      FILFDL = FILTCN * 510.
      FBLDSF = (3.*FILTAR/PI) + ((6.*FILTAR/PI)*NTFILT)
C     FIXATION PUG MILL BUILDING 15 * 25
      IF(IFIXS.EQ.1) FBLDSF = FBLDSF + 375.
      BLDMAT(7) = FBLDSF * 21.40 *RATMAT
      BLDLAB(7) = FBLDSF*20.40*RATLAB
      HPVAC = (FILTAR/400.)*50.
      HPFDRV = 2.
      HPFPMP = 10.
      HPFCCM=1.5
  502 CONTINUE
C
      ISBCAL=6
C
C     PROCEDURE FOR TANKS, FDN., AGITATORS
C     MOTOR     TANK     TANK ID
C       11       1     -SLURRY FEED TANK-SFT
C       13       2     -EFFLUENT HOLD TANK-EHT OR EFF-OXIDATION HOLD TANK
C       17       3     -POND FEED TANK-PFT
C        -       4     -THICKENER OVERFLOW SURGE TANK-TOST
C       27       5     -THICKENER FEED TANK - TFT
C       28       6     -THICKENER UNDERFLOW SURGE TANK-TUST
C                       OR FILTER FEED TANK
C        -       7     -FILTRATE SURGE TANK-FST
C       14       8     -VENTURI HOLD TANK -VHT
C       12       9     -OXIDATION-WASTE TANK  OWT
C       15      10     -RECIRCULATION HOLD TANK   OHT
C
      OTR=VTR
      IF(IFOX.EQ.1) TR=TR+OTR/4
      PI=3.1415926
      GPFT3=7.48
      DO 10000 NTK=1,10
      VOLTNK(NTK)=0.0
      DIATNK(NTK)=0.0
```

```
      HTTNK(NTK)=0.0
      UNITTG(NTK)=0.0
      TNKTOR(NTK)=0.0
      AGITHP(NTK)=0.0
      TNKFDN(NTK)=0.0
      TSQFT(NTK)=0.0
      TLSQFT(NTK)=0.0
      TNKMC(NTK)=0.0
      TNKLC(NTK)=0.0
      AGITMC(NTK)=0.0
      AGITLC(NTK)=0.0
      SPGR(NTK)=0.0
      NOTNK(NTK)=0
10000 CONTINUE
      SPGR(1)=TOTFLO(9)*7.48/(GPM(9)*60.*62.4)
      SPGR(2)=TOTFLO(11)*7.48/(GPM(11)*60.*62.4)
      SPGR(3)=TOTFLO(6)*7.48/(GPM(6)*60.*62.4)
      SPGR(5)=SPGR(3)
      IF(GPM(14).NE.0.0) SPGR(4)=TOTFLO(14)*7.48/(GPM(14)*60.*62.4)
      IF(GPM(13).NE.0.0) SPGR(6)=TOTFLO(13)*7.48/(GPM(13)*60.*62.4)
      IF(GPM(17).NE.0.0) SPGR(7)=TOTFLO(17)*7.48/(GPM(17)*60.*62.4)
      SPGR(8)=TOTFLO(12) * 7.48 / (GPM(12) * 60. *62.4)
      SPGR(9) = TOTFLO(6)*7.48/(GPM(6)*60*62.4)
      IF(GPM(19).NE.0.0) SPGR(10)=TOTFLO(19)*7.48/(GPM(19)*60*62.4)
      NOTNK(1)=1
      NOTNK(2)=NTOTAL
      NOTNK(3)=1
      SFTRT=480.
      PFTRT=5.
      OXDRT = 200
      VOLTNK(1)=GPM(9)*SFTRT*1.10
      GPMTRN=GPM(11)/NOTRAN
      VOLTNK(2)=GPMTRN*TR*1.10
      VSCTNK=VOLTNK(2)
      DO 11010 NV=1,2
      DIATNK(NV)=(4.*VOLTNK(NV)/(7.48*PI))**.3333
      HTTNK(NV)=DIATNK(NV)
      IF(HTTNK(NV).LE.60.) GO TO 11010
      HTTNK(NV)=60.
      DIATNK(NV)=SQRT(4.*VOLTNK(NV)/(7.48*PI*60.))
11010 CONTINUE
      VOLTNK(3)=GPM(6)*PFTRT*1.10
      HTTNK(3)=HTTNK(2)
      NV=3
11020 CONTINUE
      DIATNK(NV)=SQRT(4.*VOLTNK(NV)/(7.48*PI*HTTNK(NV)))
      HTODIA=HTTNK(NV)/DIATNK(NV)
      IF(HTODIA.LE.2.) GO TO 11030
      DIATNK(NV)=HTTNK(NV)/2.
      VOLTNK(NV)=PI*DIATNK(NV)*DIATNK(NV)*HTTNK(NV)*7.48/4.0
11030 CONTINUE
      IF(NV.EQ.4) GO TO 11040
```

```
      IF(ISLUDG.EQ.1) GO TO 11090
      TOSTRT=15.
      NOTNK(4)=(NOTHIK+1)/2
      VOLTNK(4)=GPM(14)*TOSTRT*1.10/NOTNK(4)
      HTTNK(4)=THKHT
      NV=4
      GO TO 11020
11040 CONTINUE
      IF(NOTHIK.EQ.1) GO TO 11050
      NOTNK(5)=1
      TFTRT=15.
      VOLTNK(5)=GPM(6)*TFTRT*1.10/NOTNK(5)
      NV=5
11050 CONTINUE
      IF(ISLUDG.LE.1.OR.(NOTHIK.LE.1.AND.ISLUDG.LT.4)) GO TO 11060
      NOTNK(6)=1
      TUSTRT=15.
      VOLTNK(6)=GPM(13)*TUSTRT*1.10/NOTNK(6)
      NV=5
11060 CONTINUE
      IF(ISLUDG.LE.3) GO TO 11070
      FSTRT=15.
      NOTNK(7)=1
      VOLTNK(7)=GPM(17)*FSTRT*1.10/NOTNK(7)
      IF(NOFILT.EQ.1) NV=7
11070 CONTINUE
      IF(ISLUDG.LE.3.AND.NOTHIK.EQ.1) GO TO 11090
      DIATNK(NV)=(4.*VOLTNK(NV)/(7.48*PI))**.3333
      HTTNK(NV)=DIATNK(NV)
      IF(HTTNK(NV).LE.60.) GO TO 11080
      HTTNK(NV)=60.
      DIATNK(NV)=SQRT(4.*VOLTNK(NV)/(7.48*PI*60.))
11080 CONTINUE
      NV=NV+1
      IF(NV.EQ.6) GO TO 11070
      IF(NV.EQ.7.AND.ISLUDG.GE.4) GO TO 11070
11090 CONTINUE
      IF(ISCRUB.LE.2)GO TO 11120
      NOTNK(8) =NTOTAL
      HTTNK(8) =HTTNK(2)
      GPMVTN=GPM(12) / NOTRAN
      VOLTNK(8) =GPMVTN * VTR * 1.10
      IF (ISCRUB.NE.4.AND.ISCRUB.NE.6) GO TO 11100
      VOLTNK(8)=VOLTNK(8)+VOLTNK(2)
      VOLTNK(2)=0.0
      DIATNK(2)=0.0
      NOTNK(2)=0
11100 CONTINUE
      DIATNK(8) =SQRT(4.*VOLTNK(8) / (7.48 * HTTNK(8) * PI))
      HTODIA =HTTNK(8) /DIATNK(8)
      IF (HTODIA.LE.2) GO TO 11110
      DIATNK(8) =HTTNK(8) / 2.0
```

```
      VOLTNK(8)=PI*DIATNK(8)*DIATNK(8)*HTTNK(8)*7.48/4.0
11110 VTR=VOLTNK(8)/(GPMVTN*1.10)
C     GO TO 294
C 293 CONTINUE
C     VOLTNK(2)=VOLTNK(2)+(GPM(12)*VTR*1.10/NOTRAN)
C     DIATNK(2)=(4.*VOLTNK(2)/(7.48*PI))**.3333
C     HTTNK(2)=DIATNK(2)
C     IF(HTTNK(2).LE.60.)GO TO 294
C     HTTNK(2)=60
C     DIATNK(2)=SQRT(4.*VOLTNK(2)/(7.48*PI*60.))
C     HTODIA=HTTNK(2)/DIATNK(2)
C     IF(HTODIA.LE.2)GO TO 294
C     DIATNK(2)=HTTNK(2)/2
C     VOLTNK(2)=PI*DIATNK(2)*DIATNK(2)*HTTNK(2)*7.48/4
C     VTR=VOLTNK(2)/GPMVTN*1.10
11120 CONTINUE
      IF(IFOX.NE.3) GO TO 11130
      NOTNK(9) = NTOTAL
      VOLTNK(9) =GPM(6) * OXDRT * 1.10
      HTTNK(9) = HTTNK(2)
      DIATNK(9) = SQRT(4*VOLTNK(9)/(7.48*HTTNK(9)*PI))
      HTODIA = HTTNK(9)/DIATNK(9)
      IF(HTODIA.LE.2) GO TO 11130
      DIATNK(9) = HTTNK(9)/2.0
      VOLTNK(9) = PI*DIATNK(9)*DIATNK(9)*HTTNK(9)*7.48/4.0
11130 CONTINUE
      IF(IFOX.NE.2.OR.ISCRUB.GE.3) GO TO 11150
      NOTNK(10) = NTOTAL
      GPMOXD = GPM(19) / NOTRAN
      IF(GPMOXD.EQ.0) GO TO 11150
      VOLTNK(10) = GPMOXD * OTR * 1.10
      HTTNK(10) = HTTNK(2)
      DIATNK(10) = SQRT(4*VOLTNK(10)/(7.48*HTTNK(10)*PI))
      HTODIA = HTTNK(10)/DIATNK(10)
      IF(HTODIA.LE.2) GO TO 11140
      DIATNK(10) = HTTNK(10)/2.0
      VOLTNK(10) = PI*DIATNK(10)*DIATNK(10)*HTTNK(10)*7.48/4.0
11140 OTR = VOLTNK(10)/(GPMOXD*1.10)
11150 CONTINUE
      PSADD=60.
      IF(IALK.EQ.2)PSADD=20.
      PCSOLD(1)=PSADD
      PCSOLD(2)=WPS
      PCSOLD(3)=WPS/2.
      PCSOLD(4)=0.0
      PCSOLD(5)=WPS
      PCSOLD(6)=PSC
      PCSOLD(7)=0.0
      PCSOLD(8)=WPS
      PCSOLD(9)=WPS
      PCSOLD(10)=WPS
      DO 1170 NTK=1,10
```

```
      IF(DIATNK(NTK).EQ.0.0) GO TO 1170
      IF(PCSOLD(NTK).EQ.0.) GO TO 1160
      IF(SPGR(NTK).EQ.0.0) GO TO 1170
      UNITTG(NTK)=0.012*PCSOLD(NTK)+0.52
      TNKTOR(NTK)=UNITTG(NTK)*VOLTNK(NTK)
      AGITHP(NTK)=0.00681*SQRT((UNITTG(NTK)**3)*(VOLTNK(NTK)
     1**3)/(SPGR(NTK)*DIATNK(NTK)**5))
 1160 CONTINUE
      TNKFDN(NTK)=.23286*DIATNK(NTK)**1.4975
      TSQFT(NTK)=3.5416*DIATNK(NTK)*HTTNK(NTK)+.7854*
     &DIATNK(NTK)*DIATNK(NTK)
      TLSQFT(NTK)=TSQFT(NTK)+.4*DIATNK(NTK)*HTTNK(NTK)
 1170 CONTINUE
      TLSQFT(4)=0
      TLSQFT(7)=0
      DO 1220 NTK=1,10
      IF(DIATNK(NTK).GT.30.) GO TO 1180
      TNKMC(NTK)=(10.2*TSQFT(NTK)*.56+TLSQFT(NTK)*4.70)*NOTNK(NTK)
      TNKLC(NTK)=(10.2*TSQFT(NTK)*.41+TLSQFT(NTK)*4.20)*NOTNK(NTK)
      GO TO 1190
 1180 TNKMC(NTK)=(12.75*TSQFT(NTK)*.56+TLSQFT(NTK)*4.70)*NOTNK(NTK)
      TNKLC(NTK)=(12.75*TSQFT(NTK)*.41+TLSQFT(NTK)*4.20)*NOTNK(NTK)
 1190 IF(TNKTOR(NTK).GT.17820.) GO TO 1200
      IF(TNKTOR(NTK).LT.1.)GO TO 1221
      AGITMC(NTK)=3900.49+TNKTOR(NTK)*0.793249*NOTNK(NTK)
      GO TO 1210
 1200 AGITMC(NTK)=45.3296*TNKTOR(NTK)**0.616464*NOTNK(NTK)
 1210 AGITLC(NTK)=.08*AGITMC(NTK)*NOTNK(NTK)
 1221 CONTINUE
 1220 CONTINUE
C
      ISBCAL=7
      IF(IFOX.LE.0) GO TO 21
      NT=2
      COMCFM = OXSCFM/NOTRAN
      IF(ISCRUB.GE.3) NT = 8
      IF(IFOX.EQ.3) NT = 9
      SPG = SPGR(NT)
      TKHEAD = HTTNK(NT)
      SPRDIA = DIATNK(NT) / 2
      NSCOMP = NOTRAN / 2
      DISPES = (TKHEAD*.85*12*62.4*SPG)/ 1728
C
C     COMPRESSOR HP , P 360 MANUAL PROCESS ENGR CAL. CLARKE-DAVIDSON
C
      COMPHP = (COMCFM * DISPES) / (229.3 * .65)
      NOCOMP = NOTRAN + NSCOMP
      SPARMC =(SPRDIA * 505.) + 2560.
      SPARLC = 5461. + 2095.
      IF(COMCFM.LT.3025.) GO TO 1002
      SPARMC = (SPRDIA * 800.) + 3650
      SPARLC = 5461. + 2650.
```

```
      IF(COMCFM.LT.4075.) GO TO 1002
      SPARMC = (SPRDIA * 990.) + 4960.
      SPARLC = 5461. + 3495.
 1002 CONTINUE
      SPARMC = SPARMC * NTOTAL
      SPARLC = SPARLC * NTOTAL
      BLDMAT(5) = 5200. * NOCOMP * RATMAT
      BLDLAB(5) = 5000. * NOCOMP * RATLAB
      COPRMC = NOCOMP * 200.0 *COMCFM** 0.63886
      COPRLC = NOCOMP * 800.
      FNO = NOCOMP * 5.
C
21    ISBCAL=8
      CALL SLPUMP
C
      ISBCAL=9
C
      EQAMAT=0.0
      EQALAB=0.0
      ADFMAT=0.0
      ADFLAB=0.0
      IF(IALK.EQ.2) GO TO 1097
C     LIMESTONE PREP. AREA SUMMATION
      EQUMAT(1)=PEQMC1
      EQUMAT(2)=PEQMC2+TNKMC(1)+AGITMC(1)+TPMPMC(1)+TPMPMC(2)
      EQULAB(1)=PEQLC1
      EQULAB(2)=PEQLC2+TNKLC(1)+AGITLC(1)+TPMPLC(1)+TPMPLC(2)
      PCONM5=(PMPFDN(1)+PMPFDN(2)+TNKFDN(1))*180.
      FDNMAT(1)=PCONM1+PCONM2
      FDNMAT(2)=PCONM4+PCONM5
      FDNLAB(1)=PCONL1+PCONL2
      FDNLAB(2)=PCONL4+(312/180*PCONM5)
      STRMAT(1)=PSTLM1+PSTLM2+PSTLM3
      STRMAT(2)=PSTLM4
      STRLAB(1)=PSTLL1+PSTLL2+PSTLL3
      STRLAB(2)=PSTLL4
      GO TO 1098
C     LIME PREP. AREA SUMMATION
 1097 CONTINUE
      EQUMAT(1)=PEQMC1
      EQUMAT(2)=TNKMC(1)+AGITMC(1)+TPMPMC(1)+TPMPMC(2)+PEQMC2
      EQULAB(1)=PEQLC1
      EQULAB(2)=TNKLC(1)+AGITLC(1)+TPMPLC(1)+TPMPLC(2)+PEQLC2
      PCONM5=(PMPFDN(1)+PMPFDN(2)+TNKFDN(1))*180.
      FDNMAT(1)=PCONM1
      FDNMAT(2)=PCONM2+PCONM5
      FDNLAB(1)=PCONL1
      FDNLAB(2)=PCONL2+(312/180*PCONM5)
      STRMAT(1)=PSTLM1
      STRMAT(2)=PSTLM2
      STRLAB(1)=PSTLL1
      STRLAB(2)=PSTLL2
```

```
 1098 CONTINUE
      IF(IADD.LT.1) GO TO 1099
C     ADDITIVE PREPARATION AREA SUMMATION
      EQAMAT = APREPM
      EQALAB = APREPL
      PCONM6 = ADMFND
      ADFMAT = PCONM6 * 180.
      ADFLAB = PCONM6 * 312.
 1099 CONTINUE
C      SUMMATION OF PREP AREA COSTS
      EQUMAT(1) = (EQUMAT(1) + EQAMAT) * RATMAT
      EQUMAT(2) = (AWFBMC+EQUMAT(2)) * RATMAT
      EQULAB(1) = (EQULAB(1) + EQALAB) * RATLAB
      EQULAB(2) = (AWFBLC + EQULAB(2)) * RATLAB
      FDNMAT(1) = (FDNMAT(1) + ADFMAT) * RATMAT
      FDNMAT(2) = (FDNMAT(2) + 0.0) * RATMAT
      FDNLAB(1) = (FDNLAB(1) + ADFLAB) * RATLAB
      FDNLAB(2) = (FDNLAB(2) + 0.0) * RATLAB
      STRMAT(1) = (STRMAT(1) + ADSMAT) * RATMAT
      STRMAT(2) = (STRMAT(2) + ASM2) * RATMAT
      STRLAB(1) = (STRLAB(1) + ADSLAB) * RATLAB
      STRLAB(2) = (STRLAB(2) + ASL2) * RATLAB
C
      CMECLC = 0.0
      CMECMC = 0.0
C
      ISBCAL=10
      IF(IESP.NE.1) GO TO 1000
      CMECMC=3572.78*ACFM**0.355025
      CMECLC=1783.61*ACFM**0.290389
 1000 CONTINUE
C
      ISBCAL=11
C
      ISBCAL=12
c      CALL FANS
C
      ISBCAL=13
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     TOP INLET DUCT TO BOTTOM GRID  =  5 FEET
C     GRID SPACING =  4 FEET
C     TOP GRID TO WASH TRAY  =  10 FEET
C     WASH TRAY TO DEMISTER  =  5 FEET
C     DEMISTER TO TOP TANGENT OF SCRUBBER  =  5 FEET
C * * * * * * * * * * * * ** * * * * * * * * * * * * * * * * * * * * * * * *
C
      ACFMV=ACFM/NOTRAN
      WRITE(8,*) '******************'
      DO 1001 M=1,10
      CSCRUB(M)=XSCRUB(M,1)*ACSTRN**XSCRUB(M,2)*NTOTAL
```

```
      WRITE(8,1776) M,CSCRUB(M),NTOTAL
 1776 FORMAT(//,' M= ',I2,' CSCRUB(M)=',G12.4)
 1001 CONTINUE
      IF (ISCRUB.EQ.1.OR.ISCRUB.EQ.3.OR.ISCRUB.EQ.4) CSCRUB(5)=CSCRUB(4)
      CSCRUB(5)=CSCRUB(5)*(LG/60)**.6
      CSCRUB(6)=CSCRUB(6)*NGRID
      CSCRUB(10)=CSCRUB(10)*(HS/9.)*(FLOAT(NSTAGE)/3.)*(82./66.)
      SCRUBM=(CSCRUB(1)+CSCRUB(2)+CSCRUB(3)+CSCRUB(5)+CSCRUB(10)
     &+CSCRUB(6))
      SCRUBL=CSCRUB(7)*19.50
      IF (ISCRUB.LT.3) GO TO 1011
      CSCRUB(11)=XSCRUB(11,1)*ACFMV**XSCRUB(11,2)*NTOTAL
      CSCRUB(12)=XSCRUB(12,1)*ACFMV**XSCRUB(12,2)*NTOTAL
      CSCRUB(13)=XSCRUB(8,1)*(ACSTRN*V/12.5)**XSCRUB(8,2)*NTOTAL
      CSCRUB(14)=XSCRUB(9,1)*(ACSTRN*V/12.5)**XSCRUB(9,2)*NTOTAL
 1011 CONTINUE
C
C
      REHTLC=0.0
      REHTMC=0.0
C
      ISBCAL=14
      IF(IRH.LE.0) GO TO 100
C
      INCONC=3449.62*ARINCL(2)**.65173
      CORTNC=462.820*(ARCORT(2))**.795585
      INCOLC=107.316*(ARINCL(2)**.692713)
      CORTLC=399.205*(ARCORT(2)**.514432)
      REHTMC=(INCONC+CORTNC)*NTOTAL
      REHTLC=(INCOLC+CORTLC)*NTOTAL
C
100   ISBCAL=15
C
C     SOOTBLOWERS  ( TWO-PRESAT.;FIVE-VENTURI;SIX-SCRUBBER) AIR-FIXED
C     FOUR-REHEAT; RETRACTABLE;   PER TRAIN
      NSBF=8*NTOTAL
      NSBR=4*NTOTAL
      IF (ISCRUB.GT.2) NSBF=11*NTOTAL
      IF(IRH.LT.1) NSBR = 0
      SOBLMC=NSBF*4000
      SOBM=NSBR*15000
      SOBLLC=NSBF*600
      SOBL=NSBR*1000
C
      ISBCAL=16
C     SUPERNATE AND MAKEUP H2O PUMPS
C      WATER PUMP          MOTOR          WATER PUMP
C       NUMBER            NUMBER             NAME
C         1                 18          MAKEUP WATER PUMP
C         2                 19          POND SUPERNATE PUMP
C         3                 20          THICKENER OVERFLOW PUMP
C         4                 21          FILTRATE PUMP PER FILTER
```

```
C         5                    22          TOTAL FILTRATE PUMP
      DO 1007 NV=1,5
      NM=17 + NV
      VERGPM(NV)=0.0
      VERHD(NV)=0.0
      VERTHP(NV)=0.0
      VERTHP(NV+3)=0.0
      VPMPMC(NV)=0.0
      VMOTMC(NV)=0.0
      VTOTMC(NV)=0.0
      VTOTLC(NV)=0.0
      NOPMOT(NM)=0
      NSPMOT(NM)=0
      NOMOT(NM)=0
 1007 CONTINUE
      VERGPM(1)=GPM(5)+GPM(7)
      GPMIN1=ACS*0.5
      IF (GPM(5).LT.ACS*0.5) GPMIN1=ACS*1.5
      IF (VERGPM(1).LT.GPMIN1) VERGPM(1)=GPMIN1
      VERHD(1)=200
      IF(ISLUDG.GT.2)GO TO 1017
      VERGPM(2)=GPM(7)
      IF(ISLUDG.EQ.2)VERGPM(2)=GPM(15)
      VERHD(2)=(0.02)*(DISTPD+300.)+80.
 1017 IF(ISLUDG.EQ.1)GO TO 1027
      VERGPM(3)=GPM(14)/NOTNK(4)
      VERHD(3)=75.
      IF(ISLUDG.LE.3)GO TO 1027
      VERGPM(4)=GPM(17)/NOFILT
      VERHD(4)=20.
      VERGPM(5)=GPM(17)/NOTNK(7)
      VERHD(5)=85.
 1027 CONTINUE
      DO 1057 NV=1,5
      NM=17 + NV
      IF(WERGPM(NV).EQ.0.)GO TO 1057
      NOPMOT(NM)=(VERGPM(NV)/10000.)+1.
      NSPMOT(NM)=1
      VERGPM(NV)=VERGPM(NV)/NOPMOT(NM)
      VERTHP(NV)=VERGPM(NV)*VERHD(NV)*1.0/(3957.*0.6)
      IF(WERGPM(NV).GT.5000.)GO TO 1037
      VPMPMC(NV)=2618.87*VERGPM(NV)**0.02392
      GO TO 1047
 1037 CONTINUE
      VPMPMC(NV)=25.2908*VERGPM(NV)**0.697554
 1047 VMOTMC(NV)=242.0 + VERTHP(NV) * 30.1766
      NOPAIR=1
      IF(NM.EQ.20) NOPAIR=NOTNK(4)
      IF(NM.EQ.21) NOPAIR=NOFILT
      IF(NM.EQ.22) NOPAIR=NOTNK(7)
      NOPMOT(NM)=NOPMOT(NM)*NOPAIR
      NSPMOT(NM)=NSPMOT(NM)*NOPAIR
```

```
      NOMOT(NM)=NOPMOT(NM)+NSPMOT(NM)
      VTOTLC(NV)=0.15*(VPMPMC(NV) + VMOTMC(NV))*NOMOT(NM)
      VTOTMC(NV)=(VPMPMC(NV) + VMOTMC(NV))*NOMOT(NM)
 1057 CONTINUE
C     STRUCTURAL STEEL  1240/TON MAT, 2200/TON LABOR
      VSTLMC=1240.*(VERTHP(2)/75)**.5*NOMOT(19)
      VSTLLC=2.2*VSTLMC
      STRMAT(7)=VSTLMC*RATMAT
      STRLAB(7)=VSTLLC*RATLAB
      TOPUMP = 0.0
      TFPUMP = 0.0
C     FOUNDATION THICK OVERFLOW & TOTAL FILTRATE PUMPS
      IF(ISLUDG.GT.1) TOPUMP = 0.25689*VERGPM(3)**0.24729*NOMOT(20)
      IF(ISLUDG.GE.4) TFPUMP = 0.25689*VERGPM(5)**0.24729*NOMOT(22)
      VERFDN = TOPUMP + TFPUMP
C
      ISBCAL=17
C
      EQUMAT(3) = FANMC*RATMAT
      EQUMAT(4)=(CMECMC+SCRUBM+VTOTMC(1)+TNKMC(8)+AGITMC(8)+CSCRUB(11)+
     &TNKMC(2)+AGITMC(2)+TPMPMC(4)+TPMPMC(5)+SOBLMC)*RATMAT
      EQUMAT(5)=(AGITMC(10)+TPMPMC(6)+COPRMC+SPARMC+TNKMC(9)+AGITMC(9)+
     & TNKMC(10))*RATMAT
      EQUMAT(6) = (REHTMC+SOBM) * RATMAT
      EQULAB(3)=FANLC*RATLAB
      EQULAB(4)=(CMECLC+SCRUBL+VTOTLC(1)+TNKLC(8)+AGITLC(8)+CSCRUB(12)+
     &TNKLC(2)+AGITLC(2)+TPMPLC(4)+TPMPLC(5)+SOBLLC)*RATLAB
      EQULAB(5)=(AGITLC(10)+TPMPLC(6)+COPRLC+SPARLC+TNKLC(9)+AGITLC(9)+
     &TNKLC(10))*RATLAB
      EQULAB(6) = (SOBL + REHTLC) * RATLAB
      DO 1016I=3,7
      J=I+4
      K=I-1
      IF(K.EQ.6) GO TO 1107
      EQUMAT(7)=EQUMAT(7)+TNKMC(I)+AGITMC(I)+TPMPMC(J)
     &+VTOTMC(K)
      EQULAB(7)=EQULAB(7)+TNKLC(I)+AGITLC(I)+TPMPLC(J)+
     &VTOTLC(K)
      GO TO 1016
 1107 CONTINUE
      EQUMAT(7)=(EQUMAT(7)+TNKMC(I)+AGITMC(I)+TPMPMC(J)+
     &TOTTKM+TFILTM+FCCTMC)*RATMAT
      EQULAB(7)=(EQULAB(7)+TNKLC(I)+AGITLC(I)+TPMPLC(J)+TOTTKL
     &+TFILTL+FCCTLC)*RATLAB
 1016 CONTINUE
C
      ISBCAL=18
c      IF (IEQPR.GE.1) CALL EQUIPR
 1020 CONTINUE
      IF(IFIXS.LT.1) GO TO 1025
      EQUMAT(7) = EQUMAT(7) + TFIXEQ
      EQULAB(7) = EQULAB(7) + TFIXLB
```

```
1025 CONTINUE
     RETURN
     END
```

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM TO DESIGN LIMESTONE PREPARATION AREA
C     REVISION DATE 83-05-12
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE LSPREP
     1       (PEQLC3, PEQLC4, PEQMC3, PEQMC4)
C
      COMMON/DUM/VERGPM(5), VERHD(5), VTOTLC(5), VTOTMC(5), VERTHP(8),
     1   AGITHP(10), AGITLC(10), AGITMC(10), DIATNK(10), HTTNK(10),
     2   TNKLC(10), TNKMC(10), VOLTNK(10), GPMPMP(11), PMPHP(11),
     3   TPMPLC(11), TPMPMC(11), TPH3, HPVAC, THKHT, CMECLC, CMECMC, CON3LG,
     4   CON8LG, DIASIL, DISPHD, FILTAR, HPMILL, HPSLAK, HTELE2, HTSILS,
     5   RAKEHP, REHTLC, REHTMC, SOBLLC, SOBLMC, SOBM, TFILTL, TFILTM, THIKAR,
     6   THKDIA, TPHSLK, TOTTKL, TOTTKM, VOLSIL, SOBL, NOTNK(10), NOMOT(45),
     7  NOPMOT(45), NSPMOT(45), NGR18, NOFILT, NOPREP, NOSILO, NOTHIK, NSFILT
      COMMON/DUMM/PMPFDN(11), TNKFDN(10), SPGR(10), BLDLAB(9), BLDMAT(9),
     1 EQULAB(9), EQUMAT(9), FDNLAB(9), FDNMAT(9), STRLAB(9), STRMAT(9),
     2 FBLDSF, FILFDL, FILFDM, HPFDRV, HPFPMP, VSCTNK, VERFDN, ELE2L, ELE2M,
     3 CSLAKL, CSLAKM, GPMILL, PCONL1, PCONL2, PCONL4, PCONM1, PCONM2, PCONM4,
     4 PEQLC1, PEQLC2, PEQMC1, PEQMC2, PSTLL1, PSTLL2, PSTLL3, PSTLL4, PSTLM1,
     5 PSTLM2, PSTLM3, PSTLM4, TPSLAK, TSLKHP, CMILLL, CMILLM, GPMTRN, NSUMPR
      COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
     1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
     2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
     3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
     4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
     5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
     6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
      COMMON/CSTNDX/ICEPYR, CEPNDX, INVYR, IREVYR, RATLAB, RATMAT, ARM, ARL
      COMMON/EQCOST/NOP(100), PEL(100), PEM(100), TOTPEL(9), TOTPEM(9)
      COMMON/FLOWS/GPM, MFLOW, WFLOW, TOTFLO
      REAL MFLOW(20,12), WFLOW(20,12), TOTFLO(20), GPM(20)
      COMMON/IND/ENGIN, ARCTEC, FLDEXP, FEES, CONT, START, CONINT, FILMAT(9)
      COMMON/LS/IALK, ISR, SR, SRADD, WPMGO, WPI, WPM, LSADD, LIMEAD, MGOADD, WMG
      REAL LSADD, LIMEAD, MGOADD
      COMMON/TRAN/NOTRAN, NOREDN, NTOTAL, PCNTRN, NORAW, NOUTIL, NOUNIT,
     1            NOPROD, NOSHIP, NSPREP, NSCOMP, NOCOMP, NOBYDU
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     NOP       DESCRIPTION                          LABOR
C      1         CAR SHAKER + HOIST                  19.10
C      2         CAR PULLER                          19.10
C      3         UNLOAD HOPPER                       19.10
C      4         VIBRATEING FEEDER                   19.50
C      5         BELT CONVEYOR                       19.50
C      6         INCLINE CONVEYOR                    19.50
C      7         UNLOAD PIT DUST COLLECTOR           19.25
```

```
C       8          UNLOAD PIT SUMP PUMP                19.50
C       9          STORAGE BELT CONVEYOR               19.50
C      10          STORAGE TRIPPER                     19.50
C      11          MOBILE EQUIPMENT SCRAPPER TRACTOR     00
C      12          MOBILE EQUIPMENT BUCKET TRACTOR       00
C      13          RECLAIM HOPPER                      19.10
C      14          RECLAIM FEEDER                      19.50
C      15          RECLAIM CONVEYOR                    19.50
C      16          RECLAIM INCLINE CONVEYOR            19.50
C      17          RECLAIM DUST COLLECTOR              19.25
C      18          RECLAIM SUMP PUMP                   19.50
C      19          RECLAIM BUCKET ELEVATOR             19.25
C      20          FEED CONVEYOR                       19.50
C      21          FEED CONVEYOR TRIPPER               19.50
C      22          FEED BIN                            19.10
C   * * * * * * * * * * * *  FEED PREPATION EQUIPMENT * * * * * *
C      23          FEED BIN WEIGH FEEDER               19.50
C      24          GYRATORY CRUSHERS                   19.50
C      25          BALL MILL DUST COLLECTORS           19.25
C      26          BALL MILL
C      27          MILLS PRODUCT TANK                  19.50
C      28          MILLS PRODUCT TANK AGITATOR
C      29          MILLS PRODUCT TANK SLURRY PUMP
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C      BEGIN BALL MILL  PROCEDURE
       DO 1 I=1,28
          PEL(I)=0.0
          PEM(I)=0.0
1      CONTINUE
       NSMILL=NSPREP
       RAWM = 0.0
       RAWL = 0.0
       TPH3=LSADD/2000.
       IF(TPH3.LT.1.0) TPH3=1.0
       TMILHP=HPTONW*WINDEX*TPH3
       HPMILL=50.
       NOMILL=1
       IF (TMILHP.LE.100.0) GO TO 1000
       IF (TMILHP.GT.100.0) NOMILL=2
       IF (TMILHP.GT.5000.0) NOMILL=(TMILHP-1.0)/2500.0+1.0
1000   CONTINUE
       HPMILL = TMILHP/NOMILL
       BALLTO = 0.158298*HPMILL**0.895265
       NTMILL=NOMILL+NSMILL
       IF(HPMILL.GT.200.)GO TO 1010
       CMILLM=((21266.4*HPMILL**0.408568)+BALLTO*510.)*NTMILL
       CMILLL=((9961.52*HPMILL**0.149065)+BALLTO*10.)*NTMILL
       GO TO 1030
1010   CMILLM=((58604.4+HPMILL*569.78)+BALLTO*510.)*NTMILL
       CMILLL=((3735.16*HPMILL**0.329414)+BALLTO*10.)*NTMILL
1030   NGR18=(TPH3-1.)/18.+1.
```

```
      NSUMPR=(TPH3-1.)/72.+1.
      IF(NSUMPR.LT.1) NSUMPR=1
      GPMILL=GPM(9)/NOMILL
      IF(TPH3.LE.18)GO TO 1050
      PEM(1)=78075.
      PEL(1)=12735.
      PEM(2)=64480.
      PEL(2)=19100.
      PEM(3)=15175.
      PEL(3)=6050.
      PEM(4)=6400.
      PEL(4)=520.
      PEM(5)=10525.
      PEL(5)=1450.
      PEM(6)=78450.
      PEL(6)=4885.
      PEM(7)=9925.
      PEL(7)=5240.
      PEM(8)=4100
      PEL(8)=770.
      PEM(9)=40625.+(NGR18-1)*26600.
      PEL(9)=NGR18*2000.
      PEM(10)=24975.
      PEL(10)=9240.
      PEM(11)=152900.
      PEL(11)=0.
      NOPMOT(1)=2
      PCONM1=74525.
      PCONL1=147320.
      PSTLM1=53000.+NGR18*17150.
      PSTLL1=12950.+NGR18*3900
      PEQMC1=0
      PEQLC1=0
      DO 1040 I=1,11
      NOP(I)=1
      PEQMC1=PEQMC1+PEM(I)
      PEQLC1=PEQLC1+PEL(I)
1040  CONTINUE
      GO TO 1060
1050  CONTINUE
      NOPMOT(1)=1
      PCONM1=0
      PCONL1=0
      PSTLM1=0
      PSTLL1=0
      PEM(12)=81900
      PEL(12)=0.
      NOP(12)=1
      PEQMC1=PEM(12)
      PEQLC1=PEL(12)
1060  NSPMOT(1)=1
      PEM(13)=NGR18*1180.
```

```
      PEL(13)=NGR18*830.
      PEM(14)=NGR18*6400.
      PEL(14)=NGR18*520.
      PEM(15)=21600.+(NGR18-1)*17040.
      PEL(15)=NGR18*1450.
      PEM(16)=55500.
      PEL(16)=3400.
      PEM(17)=6880.
      PEL(17)=2620.
      PEM(18)=NSUMPR*4100.
      PEL(18)=NSUMPR*770.
      PEM(19)=49735.
      PEL(19)=6730.
      IF(NTMILL.EQ.2)PEM(19)=58175.
      IF(NTMILL.LE.2)GO TO 1070
      PEM(20)=19320.+(NTMILL-3)*7080.
      PEL(20)=(NTMILL-2)*1450.
      NOP(20)=1
      PEM(21)=24975.
      PEL(21)=9240.
      NOP(21)=1
1070  CONTINUE
      PEM(22)=NTMILL*14075.
      PEL(22)=NTMILL*8180.
      DO 1075 I =13,22
      RAWM = RAWM + PEM(I)
      RAWL = RAWL + PEL(I)
1075  CONTINUE
      PEQLC1 = PEQLC1 + RAWL
      PEQMC1 = PEQMC1 + RAWM
      PEM(23)=NTMILL*19370.
      PEL(23)=NTMILL*785.
      PEM(24)=NTMILL*123000.
      PEL(24)=NTMILL*2175.
      PEM(25)=NTMILL*8800.
      PEL(25)=NTMILL*2620.
      PEM(26)=CMILLM
      PEL(26)=CMILLL
      PEM(27)=NTMILL*5485.
      PEL(27)=NTMILL*3910.
      PEM(28)=NTMILL*11700.
      PEL(28)=NTMILL*1200.
      NOP(13)=NGR18
      NOP(14)=NGR18
      NOP(15)=1
      NOP(16)=1
      NOP(17)=1
      NOP(18)=NSUMPR
      NOP(19)=1
      DO 1080 I=22,28
      NOP(I)=NTMILL
1080  CONTINUE
```

```
      PEQMC2=0.
      PEQLC2=0.
      DO 1090 I=23,28
      PEQMC2=PEQMC2+PEM(I)
      PEQLC2=PEQLC2+PEL(I)
1090  CONTINUE
      PCONM2=52210.+NGR18*35315.
      PCONL2=120200.+NGR18*97850.
      PSTLM2=NGR18*17155.
      PSTLL2=NGR18*3900.
      IF(NTMILL.GT.2)GO TO 1100
      PSTLM3=4340.
      PSTLL3=1620.
      GO TO 1110
1100  PSTLM3=8370.
      PSTLL3=3300.
1110  CONTINUE
      BLDAR1=31.231*HPMILL**0.5847
      PCONCT=((BLDAR1*1.5/27)+40+11+7)*NTMILL+ 8+2
      PCONM4=(PCONCT * 180.) +(3120.+100.857*HPMILL**.53508)*NTMILL
      PCONL4=(PCONCT * 312.) +(7375.+285.762*HPMILL**.53508)*NTMILL
      PSTLM4=NTMILL*20460.
      PSTLL4=NTMILL*36500.
      BLDMAT(2)=BLDAR1*(35.96+(NTMILL-1)*30.55)*RATMAT
      BLDLAB(2)=BLDAR1*(35.82+(NTMILL-1)*30.44) * RATLAB
      NOPREP=NOMILL
      RETURN
      END
```

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM TO DESIGN LIME PREPARATION AREA
C     REVISION DATE 83-05-12
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE LIMEPR
      COMMON/DUM/VERGPM(5),VERHD(5),VTOTLC(5),VTOTMC(5),VERTHP(8),
     1   AGITHP(10),AGITLC(10),AGITMC(10),DIATNK(10),HTTNK(10),
     2   TNKLC(10),TNKMC(10),VOLTNK(10),GPMPMP(11),PMPHP(11),
     3   TPMPLC(11),TPMPMC(11),TPH3,HPVAC,THKHT,CMECLC,CMECMC,CON3LG,
     4   CON8LG,DIASIL,DISPHD,FILTAR,HPMILL,HPSLAK,HTELE2,HTSILS,
     5   RAKEHP,REHTLC,REHTMC,SOBLLC,SOBLMC,SOBM,TFILTL,TFILTM,THIKAR,
     6   THKDIA,TPHSLK,TOTTKL,TOTTKM,VOLSIL,SOBL,NOTNK(10),NOMOT(45),
     7  NOPMOT(45),NSPMOT(45),NGR18,NOFILT,NOPREP,NOSILO,NOTHIK,NSFILT
      COMMON/DUMM/PMPFDN(11),TNKFDN(10),SPGR(10),BLDLAB(9),BLDMAT(9),
     1 EQULAB(9),EQUMAT(9),FDNLAB(9),FDNMAT(9),STRLAB(9),STRMAT(9),
     2 FBLDSF,FILFDL,FILFDM,HPFDRV,HPFPMP,VSCTNK,VERFDN,ELE2L,ELE2M,
     3 CSLAKL,CSLAKM,GPMILL,PCONL1,PCONL2,PCONL4,PCONM1,PCONM2,PCONM4,
     4 PEQLC1,PEQLC2,PEQMC1,PEQMC2,PSTLL1,PSTLL2,PSTLL3,PSTLL4,PSTLM1,
     5 PSTLM2,PSTLM3,PSTLM4,TPSLAK,TSLKHP,CMILLL,CMILLM,GPMTRN,NSUMPR
C
      COMMON/BIG1/BAGDLP,BAGRAT,BAGCOST,BGLIFE,PSSO2X,EBYPAS,RS,PSC,
     1 CCSA,EFFPS,ESPDLP,RESIST,SCARAT,FANHP,FANLC,FANMC,DELTAP,FILTSA,
     2 PSF,FILRAT,H,TL,VD,VE,TGASID,SKLAND,DRAINM,DRAINL,CLAYSK,WPSO3,
     3 WPSO4,WPCO3,WPINS,ASHCLN,WINDEX,PDMAX,FLOOD,SRAIR,OXAIR,OXSCFM,
     4 PRES,SO2RT,SO2RV,SO2RST,SDFEE,VPD,VT,HVCLN,HPTONW,
     5 ISLUDG,IFIXS,JSR,IVPD,IT,LON,LGMAX,ISTAND,NSBR,
     6 KEPASS,KPASO2,ICLAR,JSSO,JDIS,JSTR,JGPM,IRH,NSBF
      COMMON/CSTNDX/ICEPYR,CEPNDX,INVYR,IREVYR,RATLAB,RATMAT,ARM,ARL
      COMMON/EQCOST/NOP(100),PEL(100),PEM(100),TOTPEL(9),TOTPEM(9)
      COMMON/FLOWS/GPM,MFLOW,WFLOW,TOTFLO
      REAL MFLOW(20,12),WFLOW(20,12),TOTFLO(20),GPM(20)
      COMMON/IND/ENGIN,ARCTEC,FLDEXP,FEES,CONT,START,CONINT,FILMAT(9)
      COMMON/LS/IALK,ISR,SR,SRADD,WPMGO,WPI,WPM,LSADD,LIMEAD,MGOADD,WMG
      REAL LSADD,LIMEAD,MGOADD
      COMMON/TRAN/NOTRAN,NOREDN,NTOTAL,PCNTRN,NORAW,NOUTIL,NOUNIT,
     1           NOPROD,NOSHIP,NSPREP,NSCOMP,NOCOMP,NOBYDU
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     NOP            DESCRIPTION                    LABOR
C      1             CAR PULLER + HOIST             19.50
C      2             STORAGE SILO ELEVATOR          19.50
C      3             STORAGE BELT CONVEYOR          19.50
C      4             STORAGE CONVEYOR TRIPPER       19.50
C      5             CONCRETE STORAGE SILO
C      6             STORAGE SILO HOPPER BOTTOM
C      7             RECLAIM VIBRATING FEEDER       19.50
C      8             RECLAIM BELT CONVEYOR          19.50
C      9             FEED BIN ELEVATOR              19.50
```

```
C      10              FEED BELT CONVEYOR              19.50
C      11              FEED CONVEYOR TRIPPER           19.50
C      12              LIME FEED BIN
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C      13            BIN VIBRATING FEEDER              19.25
C      14            BIN WEIGH FEEDER                  19.25
C      15            SLAKER
C      16            SLAKER PRODUCT TANK
C      17            SLAKER PRODUCT TANK AGITATOR
C      18            LIME SYSTEM DUST COLLECTOR        19.25
C * * * * * * * *  ONCE AGAIN MATERIAL HANDLING EQUIPMENT * * * *
C      19            CAR PULLER + SHAKER               19.50
C      20            UNLOADING HOPPER
C      21            UNLOADING PIT DUST COLLECTOR      19.25
C      22            UNLOADING VIBRATING FEEDER        19.50
C      23            UNLOADING BELT CONVEYOR           19.50
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C      PROCEDURE FOR LIME SLAKER VARIABLES
       PI=3.1415926
       NSSLAK=NSPREP
       TPH3=LIMEAD/2000
       NOSLAK=1
       IF(TPH3.GT.2.) NOSLAK=2
       IF(TPH3.GT.66.) NOSLAK=3
       TPHSLK=TPH3/NOSLAK
       NTSLAK=NOSLAK+NSSLAK
       HPSLAK=2.226*TPHSLK**0.8793
       TSLKHP=HPSLAK*NOSLAK
       IF(TPHSLK.LE.5.0) CSLAKM=18567.1*EXP(TPHSLK*0.22)*NTSLAK
       IF(TPHSLK.GT.5.0) CSLAKM=32930.1*TPHSLK**0.3276*NTSLAK
       CSLAKL=6416.2*TPHSLK**0.277838*NTSLAK
       TPSLAK=TPHSLK
C      PROCEDURE FOR PREP. AREA EQUIP. COSTS
       VOLSTR=LIMEAD*24*30/55
C      VOLSTR=LIMEAD*6   DENSITY CHANGE & 30 DAY STORAGE
       NOSILO=(VOLSTR-1.)/147200.+1.
       VOLSIL=VOLSTR/NOSILO
C      HEIGHT OF SILO STORAGE SECTION EQUALS 1.5 * DIAMETER
       DIASIL=(8.*VOLSIL/(PI*3.))**0.333333
       RADSIL=DIASIL/2.
       HTHOP=0.866*DIASIL
       HOPLB=(PI*RADSIL*SQRT(RADSIL**2+HTHOP**2)+PI*(DIASIL/2.+.25))*10.2
       HTSILS=1.5*DIASIL
       THTSIL=HTSILS+HTHOP+5.
C OLD  HTELE2=THTSIL+5.
       HTELE2=THTSIL+5.+10.
       CON8LG=NOSILO*DIASIL+75.
       YD3SIL=PI*(DIASIL+1.)*THTSIL/27.
       FDNSIL=PI*((DIASIL+2.)/2.)**2*1.5/27.
       SILOM=YD3SIL*180.+7.664*DIASIL**1.917
       SILOL=YD3SIL*312.+11.26*DIASIL**1.98
       ELE2M=(HTELE2/90.)**0.83*40900.
```

```
      ELE2L=0.08*ELE2M
      STL2M=(HTELE2/90.)**0.83*4590.
      STL2L=0.28*STL2M
      PEM(1)=78075.
      PEL(1)=12735.
      NOP(1)=1
      PEM(2)=ELE2M
      PEL(2)=ELE2L
      NOP(2)=1
      IF(NOSILO.EQ.2)PEM(2)=ELE2M+8440.
      IF(NOSILO.LT.3)GO TO  000
      CON3LG=DIASIL*NOSILO
      PEM(3)=CON3LG*235+12550
      PEL(3)=CON3LG*66+1320
      NOP(3)=1
      PEM(4)=17475.
      PEL(4)=6600.
      NOP(4)=1
 1000 CONTINUE
      PEM(5)=NOSILO*SILOM
      PEL(5)=NOSILO*SILOL
      NOP(5)=NOSILO
      PEM(6)=NOSILO*HOPLB*0.56
      PEL(6)=NOSILO*HOPLB*0.41
      NOP(6)=NOSILO
      PEM(7)=NOSILO*4470.
      PEL(7)=NOSILO*400.
      NOP(7)=NOSILO
      PEM(8)=CON8LG*125+9650.
      PEL(8)=CON8LG*20+985.
      NOP(8)=1
      PEM(9)=32475.
      PEL(9)=2250.
      IF(NTSLAK.EQ.2)PEM(9)=40915.
      NOP(9)=1
      IF(NTSLAK.LT.3)GO TO 1010
      PREP = (NTSLAK - 1) * 30.
      PEM(10)=7930. + 122. * PREP
      PEL(10)=660. + 20. * PREP
      NOP(10)=1
      PEM(11)=18175
      PEL(11)=6600.
      NOP(11)=1
 1010 CONTINUE
      PEM(12)=NTSLAK*4710.
      PEL(12)=NTSLAK*3390.
      PEM(13)=NTSLAK*11920.
      PEL(13)=NTSLAK*1310.
      PEM(14)=NTSLAK*6110.
      PEL(14)=NTSLAK*785.
      PEM(15)=CSLAKM
      PEL(15)=CSLAKL
```

```
      PEM(16)=NTSLAK*6170.
      PEL(16)=NTSLAK*4470.
      PEM(17)=NTSLAK*8000.
      PEL(17)=NTSLAK*800.
      DO 1020 I=12,17
      NOP(I)=NTSLAK
 1020 CONTINUE
      PEM(18)=6800.
      PEL(18)=2620.
      NOP(18)=1
      PEM(19)=64480.
      PEL(19)=19100.
      NOP(19)=1
      PEM(20)=15150.
      PEL(20)=5980.
      NOP(20)=1
      PEM(21)=9925.
      PEL(21)=5280.
      NOP(21)=1
      PEM(22)=4470.
      PEL(22)=265.
      NOP(22)=1
      PEM(23)=10980.
      PEL(23)=1450.
      NOP(23)=1
      PEQMC1=0.
      PEQLC1=0.
      DO 1030 I=1,12
      PEQMC1=PEQMC1+PEM(I)
      PEQLC1=PEQLC1+PEL(I)
 1030 CONTINUE
      DO 1032 I =18,23
      PEQLC1 = PEQLC1 + PEL(I)
      PEQMC1 = PEQMC1 + PEM(I)
 1032 CONTINUE
      DO 1035 I = 13,17
      PEQLC2 = PEQLC2 + PEL(I)
      PEQMC2 = PEQMC2 + PEM(I)
 1035 CONTINUE
      PCONM1=2950.+FDNSIL*180.*NOSILO
      PCONL1=5150.+FDNSIL*312.*NOSILO
      PSTLM1=41725.+STL2M+CON8LG*58+1750.
      PSTLL1=8410.+STL2L+CON8LG*23.88+710.
      IF(NOSILO.LT.3)GO TO 1040
      PSTLM1=PSTLM1+CON3LG*73+8930.
      PSTLL1=PSTLL1+CON3LG*29.05+1165
 1040 CONTINUE
      PCONM2=1320*NTSLAK
      PCONL2=2275*NTSLAK
      PSTLM2=4960+3500*NTSLAK
      PSTLL2=1300+4725*NTSLAK
      IF(NTSLAK.LT.3)GO TO 1050
```

```
      PSTLM2=PSTLM2+12150.
      PSTLL2=PSTLL2+4335.
 1050 CONTINUE
C     W*L*H, 2 O.DOOR     30*70*30
      BLDMAT(1)=31200.*RATMAT
      BLDLAB(1)=12480.*RATLAB
      NOPREP=NOSLAK
      RETURN
      END
```

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM TO DESIGN ADDITIVE EQUIPMENT
C     REVISION DATE 83-03-25
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE ADAMGO
     1          (BLDLAB, BLDMAT)
C
      COMMON/ADDPRE/ADDSIL, DIAADD, HTSADD, IANGLE, ASILMC, ASILLC, PCOVMC,
     1PCOVLC, PCOVHP, ABINMC, ABINLC, AWFBMC, AWFBLC, ADUSTM, ADUSTL
      COMMON/ADDSUM/APREPM, APREPL, ADMFND, ADSMAT, ADSLAB, ASM2, ASL2
      COMMON/CSTNDX/ICEPYR, CEPNDX, INVYR, IREVYR, RATLAB, RATMAT, ARM, ARL
      COMMON/LIQ2/IW4, AD, ADDC, ADLOSS, AMGLOS
      COMMON/LS/IALK, ISR, SR, SRADD, WPMGO, WPI, WPM, LSADD, LIMEAD, MGOADD, WMG
      REAL LSADD, LIMEAD, MGOADD
      COMMON/SCRUB/ISCRUB, IFOX, IADD, VLG, VTR, IFAN
C
      DIMENSION BLDLAB(9), BLDMAT(9)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     NOP             DESCRIPTION
C      31             PNEUMATIC CONVEYOR SYSTEM
C      32             ADDITION FEED BIN
C      33             MGO OR ADIPIC ACID STORAGE SILO
C      34           * FAN NOT IN THIS AREA * * *
C      35             ADDITIVE DUST COLLECTOR
C  * * * * * * * * ADDITIVE PREPARATION* * * * * * * * * * * * * * * * * * *
C      30             SCREW FEEDER
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      PI = 3.1415926
      IANGLE = 60
      ADDITT = ADLOSS
      IF(IADD.NE.1) GO TO 1000
      IF(IALK.EQ.1) AMGLOS=LSADD
      IF(IALK.EQ.2) AMGLOS=LIMEAD
      AMGLOS=AMGLOS*MGOADD/100
      ADDITT=AMGLOS
 1000 CONTINUE
      ADDSIL = 720/49 * ADDITT
      DIAADD = (8 * ADDSIL / (PI * 3.) ) ** 0.33333
      HOPPER = DIAADD * .8333
      HTSADD = (1.5 * DIAADD) + HOPPER
      ASILMC = 733.70 * ADDITT ** 0.5727
      ASILLC = 449.96 * ADDITT ** 0.5988
      IF(ADDITT.GT.1000) GO TO 1010
      PCOVMC = 8250.
      PCOVLC = 5330.
      PCOVHP = 10.
      GO TO 1020
```

```
 1010 PCOVMC = 9460.
      PCOVLC = 5590.
      PCOVHP = 15.
      IANGLE = 45
C     HOPPER = DIAADD * .5
      HTSADD = 2 * DIAADD
 1020 CONTINUE
      ABINMC = 333.44 * ADDITT ** 0.45888
      ABINLC = 124.06 * ADDITT ** 0.53497
      ADUSTM = 2935.
      ADUSTL = 265.
      AWFBMC = 5500.
      AWFBLC = 525.
      ABUILM = 5400. + 3.117 * ADDITT
      IF(IALK.EQ.2) ABUILM = 0.0
      ABUILL = ABUILM * RATLAB
      BLDLAB(1) = BLDLAB(1) + ABUILL*1.2
      BLDMAT(1) = BLDMAT(1) +(ABUILM *1.3* RATMAT)
      APREPL = ASILLC + ABINLC + PCOVLC + ADUSTL
      APREPM = ASILMC + ABINMC + PCOVMC + ADUSTM
C     CONCRETE  180 M, 312 L ,1983  UNIT HERE IS CUYD
      ADMFND = 2.83 * ADDITT ** 0.536
C      STRUCTURAL STEEL  M=0.62, L=1.10
      ASTEEL = 9370. + ADDITT * 22.62
      ASM2 = 0.25 * ADDITT * 0.62
      ASL2 = 0.25* ADDITT * 1.10
      ADSMAT = ASTEEL * 0.62
      ADSLAB = ASTEEL * 1.10
      RETURN
      END
```

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM TO DESIGN SLURRY PUMPS
C     REVISION DATE 83-04-11
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE SLPUMP
      COMMON/DUM/VERGPM(5),VERHD(5),VTOTLC(5),VTOTMC(5),VERTHP(8),
     1   AGITHP(10),AGITLC(10),AGITMC(10),DIATNK(10),HTTNK(10),
     2   TNKLC(10),TNKMC(10),VOLTNK(10),GPMPMP(11),PMPHP(11),
     3   TPMPLC(11),TPMPMC(11),TPH3,HPVAC,THKHT,CMECLC,CMECMC,CON3LG,
     4   CON8LG,DIASIL,DISPHD,FILTAR,HPMILL,HPSLAK,HTELE2,HTSILS,
     5   RAKEHP,REHTLC,REHTMC,SOBLLC,SOBLMC,SOBM,TFILTL,TFILTM,THIKAR,
     6   THKDIA,TPHSLK,TOTTKL,TOTTKM,VOLSIL,SOBL,NOTNK(10),NOMOT(45),
     7  NOPMOT(45),NSPMOT(45),NGR18,NOFILT,NOPREP,NOSILO,NOTHIK,NSFILT
      COMMON/DUMM/PMPFDN(11),TNKFDN(10),SPGR(10),BLDLAB(9),BLDMAT(9),
     1 EQULAB(9),EQUMAT(9),FDNLAB(9),FDNMAT(9),STRLAB(9),STRMAT(9),
     2 FBLDSF,FILFDL,FILFDM,HPFDRV,HPFPMP,VSCTNK,VERFDN,ELE2L,ELE2M,
     3 CSLAKL,CSLAKM,GPMILL,PCONL1,PCONL2,PCONL4,PCONM1,PCONM2,PCONM4,
     4 PEQLC1,PEQLC2,PEQMC1,PEQMC2,PSTLL1,PSTLL2,PSTLL3,PSTLL4,PSTLM1,
     5 PSTLM2,PSTLM3,PSTLM4,TPSLAK,TSLKHP,CMILLL,CMILLM,GPMTRN,NSUMPR
C
      COMMON/BIG1/BAGDLP,BAGRAT,BAGCOST,BGLIFE,PSSO2X,EBYPAS,RS,PSC,
     1 CCSA,EFFPS,ESPDLP,RESIST,SCARAT,FANHP,FANLC,FANMC,DELTAP,FILTSA,
     2 PSF,FILRAT,H,TL,VD,VE,TGASID,SKLAND,DRAINM,DRAINL,CLAYSK,WPSO3,
     3 WPSO4,WPCO3,WPINS,ASHCLN,WINDEX,PDMAX,FLOOD,SRAIR,OXAIR,OXSCFM,
     4 PRES,SO2RT,SO2RV,SO2RST,SDFEE,VPD,VT,HVCLN,HPTONW,
     5 ISLUDG,IFIXS,JSR,IVPD,IT,LON,LGMAX,ISTAND,NSBR,
     6 KEPASS,KPASO2,ICLAR,JSSO,JDIS,JSTR,JGPM,IRH,NSBF
      COMMON/FLOWS/GPM,MFLOW,WFLOW,TOTFLO
      REAL MFLOW(20,12),WFLOW(20,12),TOTFLO(20),GPM(20)
      COMMON/PND1/PDEPTH,PMXEXC,DISTPD,ILINER,CLAYIN,CLAYUC,HYPLMC,
     1            HYPLLC,PSAMAX,ACREC,APRAIN,SAPILE
      COMMON/SCRUB/ISCRUB,IFOX,IADD,VLG,VTR,IFAN
      COMMON/TRAN/NOTRAN,NOREDN,NTOTAL,PCNTRN,NORAW,NOUTIL,NOUNIT,
     1            NOPROD,NOSHIP,NSPREP,NSCOMP,NOCOMP,NOBYDU
C
      DIMENSION COST(7),RHP(10),GPMSP(9),PERPMP(9)
      DIMENSION NSLPMP(11),PMPMC(11)
      DIMENSION ELECT(10,7,2)
      DATA ELECT/355.,370.,524.,2*762.,2*1672.,2*3785.,6375.,
     &           0.20,0.27,0.42,6*0.89,0.00205602,
     &           0.92,0.96,1.02,6*1.10,0.00514005,
     &           0.72,0.86,1.16,6*1.86,0.0076956,
     &           7*1.,2.,2*3.,
     &           190.,300.,510.,900.,2*1790.,3*70.7362,471.027,
     &           5*186.311,3*0.27048,2*204.728,
     &           10*0.0,
     &           9*0.0,1.00048,
     &           9*0.0,1.00048,
     &           9*0.0,0.948743,
```

```
     &           10*0.0,
     &           6*0.0,3*0.89287,0.639446,
     &           5*0.33909,3*1.80974,2*0.855266/
      DATA RHP/2.5,7.5,15.,35.,50.,90.,133.,233.,250.,1500./
      DATA GPMSP/115.,290.,450.,870.,1160.,2650.,4250.,7000.,20001./
      DATA PERPMP/1775.,2820.,4018.,4457.,7410.,8178.,13483.,21412.,
     &29968./
C
C     PROCEDURE FOR SLURRY PUMPS
C
C     SLURRY PUMP       MOTOR
C       NUMBER          NUMBER          NAME
C          1              3          MILLS PRODUCT PUMP OR SLAKER PUMP
C          2              5          SLURRY FEED PUMP
C          3              6          ADDITIVE PUMP   FUTURE
C          4              7          COOLING SLURRY OR VENTURI PUMP
C          5              8          SCRUBBER RECYCLE PUMP
C          6              9          OXIDATION BLEED
C          7             10          POND FEED PUMP
C          8             23          FEED TO THICKENER FEED PUMP
C          9             24          THICKENER FEED PUMP
C         10             25          THICKENER UNDERFLOW PUMP
C         11             26          FILTER FEED PUMP
C
C     MILLS PRODUCT PUMPS  NOPREP=OPERATING, NSPREP=SPARE
C
      NTPREP=NOPREP+NSPREP
      DO 1000 NP=1,11
      GPMPMP(NP)=0.
      NSLPMP(NP)=0.
      PMPHP(NP)=0.
      PMPFDN(NP)=0.
      PMPMC(NP)=0.
      TPMPMC(NP)=0.
      TPMPLC(NP)=0.
 1000 CONTINUE
      GPMPMP(1)=GPM(9)/NOPREP
      NOPMOT(3)=NOPREP
      NSPMOT(3)=NSPREP
      IF(NSPMOT(3).EQ.0) NSPMOT(3)=1
      NSLPMP(1)=NOPMOT(3)+NSPMOT(3)
      PMPHP(1)=GPMPMP(1)*60.*SPGR(1)/(3957.*.6)
      GPMPMP(2)=GPM(9)/NOTRAN
      NSPMOT(5)=(NTOTAL+1)/2+NOREDN
      NOPMOT(5)=NOTRAN
      NSLPMP(2)=NSPMOT(5)+NOPMOT(5)
      PMPHP(2)=GPMPMP(2)*60.*SPGR(1)/(3957.*.6)
C     IF(IADD.LT.1) GO TO 620
C     NOPMOT(6)=1
C     NSPMOT(6)=1
C     GPMPMP(3)=GPM(18)
C     PMPHP(3)=GPMPMP(3)*60.  *SPGR(9)/(3957.*.6)
```

```
C      NSLPMP(3)=NOPMOT(6)+NSPMOT(6)
C 620 CONTINUE
      NOPMOT(7)=NOTRAN
      NSPMOT(7)=NTOTAL+NOREDN
      IF (GPM(12)/NOTRAN.LE.20000) GO TO 1010
      NOPMOT(7)=2*NOTRAN
      NSPMOT(7)=NTOTAL+2*NOREDN
 1010 GPMPMP(4)=GPM(12)/NOPMOT(7)
      NSLPMP(4)=NSPMOT(7)+NOPMOT(7)
      PMPHP(4)=GPMPMP(4)*100.*SPGR(2)/(3957.*.6)
      IF(ISCRUB.GE.3) PMPHP(4)=GPMPMP(4)*100.*SPGR(8)/(3957.*.6)
      NOPMOT(8) = 2 * NOTRAN
      NSPMOT(8) = NTOTAL + 2*NOREDN
      IF(GPMTRN.LT.40000.) GO TO 1020
      NOPMOT(8) = 3 * NOTRAN
      NSPMOT(8) = NTOTAL + 3 * NOREDN
      IF(GPMTRN.LT.60000.) GO TO 1020
      NOPMOT(8) = 4 * NOTRAN
      NSPMOT(8) = NTOTAL + 4 * NOREDN
      IF(GPMTRN.LT.80000.) GO TO 1020
      NOPMOT(8) = 5 * NOTRAN
      NSPMOT(8) = NTOTAL + 5 * NOREDN
 1020 CONTINUE
      NSLPMP(5)=NSPMOT(8)+NOPMOT(8)
      GPMPMP(5)=GPM(11)/NOPMOT(8)
      PMPHP(5)=GPMPMP(5)*100.*SPGR(2)/(3957.*.6)
      IF(IFOX.EQ.0) GO TO 1030
      NOPMOT(9) = NOTRAN
      IF(GPM(6) .GT. 20000.) NOPMOT(9) = 2 * NOTRAN
      NSPMOT(9) = (NTOTAL + 1) / 2 + NOREDN
      NSLPMP(6) = NSPMOT(9) + NOPMOT(9)
      GPMPMP(6) = GPM(6) / NOPMOT(9)
      PMPHP(6) = GPMPMP(6) * 60. * SPGR(10)/(3957.*0.6)
 1030 CONTINUE
      IF(ISLUDG.GT.2) GO TO 1070
      GPMP5=GPM(6)
      SPGRP5=SPGR(3)
      IF (ISLUDG.EQ.1) GO TO 1040
      GPMP5=GPM(13)
      SPGRP5=SPGR(6)
 1040 CONTINUE
      DISPHD=0.02*(DISTPD+200.)+20.
      IF(GPMP5.GE.255.) GO TO 1050
      NOPMOT(10)=2
      NSPMOT(10)=2
      PMPHP(7)=GPMP5*(DISPHD/2.0)*SPGRP5/(3957.0*0.6)
      GPMPMP(7)=GPMP5
      GO TO 1060
 1050 CONTINUE
      NOPMOT(10)=1
      IF(GPMP5.GT.20000.)  NOPMOT(10)=2
      PMPHP(7)=GPMP5*DISPHD*SPGRP5/(3957.0*0.6*FLOAT(NOPMOT(10)))
```

```
      NSPMOT(10)=1
      GPMPMP(7)=GPMP5/NOPMOT(10)
 1060 NSLPMP(7)=NOPMOT(10)+NSPMOT(10)
      IF(ISLUDG.EQ.1) GO TO 1120
 1070 CONTINUE
      IF(NOTHIK.EQ.1) GO TO 1080
      NOPMOT (23)=1
      IF(GPM(6).GT.20000.)  NOPMOT(23)=2
      PMPHP(8)=GPM(6)*60.*SPGR(3)/(3957.*.6*NOPMOT(23))
      NSPMOT(23)=1
      NSLPMP(8)=NOPMOT(23)+NSPMOT(23)
      GPMPMP(8)=GPM(6)/NOPMOT(23)
 1080 CONTINUE
      NOPMOT(24)=NOTHIK
      NSPMOT(24)=(NOTHIK+1)/2
      NSLPMP(9)=NOPMOT(24)+NSPMOT(24)
      GPMPMP(9)=GPM(6)/NOPMOT(24)
      PMPHP(9)=GPMPMP(9)*60.*SPGR(3)/(3957.*.6)
      IF(NOTHIK.LE.1.AND.ISLUDG.LT.4) GO TO 1090
      NOPMOT(25)=NOTHIK
      NSPMOT(25)=NOTHIK
      NSLPMP(10)=NOPMOT(25)+NSPMOT(25)
      GPMPMP(10)=GPM(13)/NOPMOT(25)
      PMPHP(10)=GPMPMP(10)*HTTNK(6)*SPGR(6)/(3957.*.6)
 1090 CONTINUE
      IF(ISLUDG.EQ.2) GO TO 1120
      IF(ISLUDG.GT.3) GO TO 1100
      NOPMOT(26)=1
      IF(GPM(13).GT.20000.)NOPMOT(26)=2
      NSPMOT(26)=1
      GO TO 1110
 1100 CONTINUE
      NOPMOT(26)=NOFILT
      NSPMOT(26)=(NOFILT+1)/2
 1110 CONTINUE
      NSLPMP(11)=NOPMOT(26)+NSPMOT(26)
      GPMPMP(11)=GPM(13)/NOPMOT(26)
      PMPHP(11)=GPMPMP(11)*50.*SPGR(6)/(3957.*.6)
 1120 CONTINUE
      DO 1220 NP=1,11
      DO 1130 NG=1,9
      IF(GPMPMP(NP).LE.GPMSP(NG))GO TO 1140
 1130 CONTINUE
      NG=9
 1140 PMPMC(NP)=PERPMP(NG)*NSLPMP(NP)
      IF(NG.GT.1)GO TO 1150
      TPMPLC(NP)=900.*NSLPMP(NP)
      GO TO 1160
 1150 TPMPLC(NP)=135.076*GPMPMP(NP)**0.437034*NSLPMP(NP)
 1160 IF(GPMPMP(NP).GT.3200.)GO TO 1170
      PMPFDN(NP)=.25689*GPMPMP(NP)**.24729*NSLPMP(NP)
      GO TO 1180
```

```
1170 PMPFDN(NP)=.84439*GPMPMP(NP)**.19138*NSLPMP(NP)
1180 DO 1190 IN=1,10
     IF(PMPHP(NP).LE.RHP(IN))GO TO 1200
1190 CONTINUE
     IN=10
1200 II=IN
     DO 1210 JJ=6,7
     IF(PMPHP(NP).EQ.0.0) GO TO 1205
     COST(JJ)=ELECT(II,JJ,1)*PMPHP(NP)**ELECT(II,JJ,2)*NSLPMP(NP)
     GO TO 1210
1205 COST(JJ) = 0.0
1210 CONTINUE
     TPMPMC(NP)=PMPMC(NP)+COST(6)+COST(7)
1220 CONTINUE
     NOMOT(3)=NSLPMP(1)
     DO 1230 I=2,11
     J=I+3
     IF (I.GT.7) J=I+15
     NOMOT(J)=NSLPMP(I)
1230 CONTINUE
     RETURN
     END
```

```
$DEBUG
C
C
C      MAIN OUTPUT DRIVER PROGRAM
C           HANDLES FANS,  WASTE DISPOSAL,  AND ECONOMIC OPTIONS
C
C           BY JEFF WINGET
C              3/22/85
C

      CHARACTER*4 HMOD,HSTRM,HCAP,SHCAP,OPTMZ,YES,INTER,DBGT
      CHARACTER*13 NEXTFL,END,BLANK1,DBGF
      CHARACTER*80 COMM1,COMM2
      INTEGER FMB,CT,BOTTOM,TYPCT,CLNLVL,FMAAHX
      LOGICAL KO(2),TVA,FGD,LOG,XXX
      REAL INDCST,NTRAIN
      DIMENSION HMOD(9),HSTRM(5,8),NST(9),HCAP(10,9),SHCAP(6,3,9),IC(10)
      INTEGER*2 CW
      INTEGER R100,AC(13),TDIR,CC(9,4),TOTALA,OVHD,OVHDC,FYAC,TALEV,
     &        TIND,TOTALC,CAPCHG,ARLEV
      COMMON /BATCHC/OPTMZ,NOPT,TARGET,NEXTFL
      COMMON /OUTH/ HMOD,HSTRM,HCAP,SHCAP,IC,NST
      COMMON /OPT/     BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
      COMMON /NEW/ IFACT,V2,FC
      COMMON /COSTS/ CAPCST(10,8),ANN(25),INDCST(10)
      COMMON /COAL1/ HHV,PS,PA,CST,PNA,ALK,CL,FEO,PM
      COMMON /STREAM/ PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
      COMMON /INPUT/BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,HHVCLN,PSCLN,PACLN,CSTCLN,PNACLN,ALKCLN,
     &CLCLN,FEOCLN,PMCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,IAPCS(9),KO
     &,TVA
      COMMON /INSYSP/ CCEFF,UTEFF,BNHR,FANLOS,QIN,ALKIN
      COMMON /PARM/ P(500)
      COMMON /COMM/ COMM1,COMM2
      COMMON /MATBAL/ TABLE(8,8)
      COMMON /FAN/ DELTAP,CSTFAN
      COMMON/BIG1/BAGDLP,BAGRAT,BAGCOST,BGLIFE,PSSO2X,EBYPAS,RS,PSC,
     1 CCSA,EFFPS,ESPDLP,RESIST,SCARAT,FANHP,FANLC,FANMC,DELTA1,FILTSA,
     2 PSF,FILRAT,H,TL,VD,VE,TGASID,SKLAND,DRAINM,DRAINL,CLAYSK,WPSO3,
     3 WPSO4,WPCO3,WPINS,ASHCLN,WINDEX,PDMAX,FLOOD,SRAIR,OXAIR,OXSCFM,
     4 PRES,SO2RT,SO2RV,SO2RST,SDFEE,VPD1,VT,HVCLN,HPTONW,
     5 ISLUDG,IFIXS,JSR,IVPD,IT,LON,LGMAX,ISTAND,NSBR,
     6 KPASS,KPASO2,ICLAR,JSSO,JDIS,JSTR,JGPM,IRH,NSBF
      COMMON/COAL/HVC,WPC,WPH,WPO,WPN,WPSUL,WPCL,WPASH,WPH2O,SULO,ASHO
      COMMON/CSTNDX/ICEPYR,CEPNDX,INVYR,IREVYR,RATLAB,RATMAT,ARM,ARL
      COMMON/FLYASH/IASH,ASHUPS,ASHSCR,ASHPRU,ASHPRS,ASHLBU,ASHLBS
      COMMON/HGAS2/MPCT,LBMH,LBH,SCFM,HHG,HWG,CFR,BCFR,RCFR
      COMMON/PCOST/POND(9),PDACRE,TOPHRS,PNDCAP,PNDLAB(9),PNDMAT(9),ELAN
      COMMON/PND1/PDEPTH,PMXEXC,DISTPD,ILINER,CLAYIN,CLAYUC,HYPLMC,
     1            HYPLLC,PSAMAX,ACREC,APRAIN,SAPILE
      COMMON/SCRUBR/LG,V,PPMOUT,SO2R,ELIMIT,PR,PD1,WPS,OX,PH,NSTAGE,
     1              NGRID,HS,ACS,ACSTRN
```

```
      COMMON/SOLDIS/CASO3, CASO4, CACO3, INSOLS, WDIS, PSD, DENS, DEND
      COMMON/OPPSCH/IOPSCH, IYROP, IA(50)
      REAL MPCT(6), LBMH(6), LBH(6)
      COMMON/LANDIS/FILAND, TWLF, DISFUL, ELECLF, WATERF, FILANL, OVLF
      DIMENSION SW(4), AREA(10), B(5), ULFC(7)
      DIMENSION B1(5)
      REAL LG
      REAL INSOLS
      COMMON/COMPLE/ICOMPL, IRPT, IRUN, NEXT, IEQPR, IWTBAL, ITM, INPOPT, NOPART
      COMMON/PCOSTS/PCLEAR, PDEXEC, SODM, SODL, PERIML, PERIMM, ROADMM, ROADLL,
     1 RECLAM
      COMMON/PNDARW/AB, AID, AIE, AOD, AS, DL, P1, VEC, VETT, WELLM, WELLL, WPSITE
      COMMON/PNDIND/INDPND, PENGIN, PARCH, PFLDEX, PFEES, PCONT, PSTART, PCONIN
      COMMON/RRIND/XINT, PCTMNT, PDMNTP, XINFLA, PCTOVR, PCTADM, UNDCAP,
     1         PCTINS, UC, IECON
      REAL *8 XINT, PCTMNT, PDMNTP, XINFLA, PCTOVR, PCTADM, UNDCAP, PCTINS, UC(1
     1       8)
      DIMENSION CYD(7), TYD(7), FUELDT(7), BYD(5)
      COMMON/TAXES/ITAXFR, TXFRRT, TXRAT, FRRAT, SERVRT, ROYALT, IOTIME, OTRATE
      COMMON /TVAFAN/ TGAS, TWG, VPD, NTRAIN, DLTAPF, NOTRAN, NTOTAL, ISCRUB,
     & NOREDN, PD, DPRH
      COMMON /ESC/ IYC, IMC, IYA, IMA, RATPLT, RATLA2, RATMA2, RATOM
      DATA YES/'YES'/
      DATA INTER/'I'/
      DATA END/'END        '/
      DATA BLANK1/'                '/
C
      OPEN (17, FILE='PASS2.BIN', STATUS='OLD', FORM='BINARY')
      OPEN (1, FILE='NEXTRUN.BIN', STATUS='NEW', FORM='BINARY')
      OPEN (55, FILE='FAILURE', STATUS='NEW')
      open (7, file='lpt1:')
      open (6, file='con:')
C
      CALL ZERO
C
      READ (17) ACREC, ASHO, ASHPRU, CLAYIN, CLAYUC, DISTPD, HYPLLC, HYPLMC
      READ (17) IA, ILINER, IRPT, ISLUDG, ITAXFR, OX, PARCH, PCLEAR, PCONIN
      READ (17) PCONT, PDEPTH, PDEXEC, PENGIN, PERIML, PERIMM, PFEES, PFLDEX
      READ (17) PMXEXC, PNDCAP, PSD, PSTART, RATLAB, RATMAT, RCFR, RECLAM
      READ (17) ROADLL, ROADMM, SODL, SODM, TOPHRS, TXFRRT, UC, WDIS, WELLL
      READ (17) WELLM, WPASH, IFIXS
C
C
      READ (17) TGAS, TWG, VPD, NTRAIN, DLTAPF, NOTRAN, NTOTAL, ISCRUB, NOREDN
      READ (17) PD, DPRH
C
      READ (17) CAPCST, INDCST, ANN
      READ (17) HHV, PS, PA, CST, PNA, ALK, CL, FEO, PM
      READ (17) PPH, ACFM, TEMP, PPHMST, PPHALK, PPHPRT, PPHSO2, PPHNOX
      READ (17) BSIZE, CF, AHFX, AHXT, HHVROM, PSROM, PAROM, CSTROM, PNAROM,
     &ALKROM, CLROM, FEOROM, PMROM, HHVCLN, PSCLN, PACLN, CSTCLN, PNACLN, ALKCLN,
     &CLCLN, FEOCLN, PMCLN, FMB, NR, IS, CT, TYPCT, CLNLVL, BOTTOM, NM,
```

```
      &IAPCS,KO,TVA
       READ (17) CCEFF,UTEFF,BNHR,FANLOS,QIN,ALKIN
       READ (17) P
       READ (17) COMM1,COMM2
       READ (17) TABLE
       READ (17) DELTAP,CSTFAN
       READ (17) IYC,IMC,IYA,IMA,RATPLT,RATLA2,RATMA2,RATOM
       READ(17) IFACT,V2,FC
       READ(17) BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
       READ (17) OPTMZ,NOPT,TARGET,NEXTFL
       CLOSE (17)
       READ(COMM1,'(A4)') DBGT
       IF(DBGT .NE. 'TEST') THEN
           DBGF= 'NUL           '
       ELSE
           DBGF= 'OUTOUT.DBG    '
       ENDIF
       open (8,file=DBGF,STATUS='NEW')
C
C
C
C
       FGD= .FALSE.
       DO 3 I=1,NM
     3 IF(IAPCS(I) .EQ. 8) FGD= .TRUE.
       WRITE(8,*) ' CALLING FAN1...'
       IF(DELTAP .NE. 0. .OR. FGD) CALL FAN1
       WRITE(8,*) ' CALLING WASTE...'
C
       IF(WDIS+ANN(13) .GT. 0.0) CALL WASTE
C
C      CALL ECONOM
C
       IF (TVA) THEN
          CALL OUTTVA
         ELSE
          CALL OUTEPR
       ENDIF
       IF(KO(2)) WRITE(7,'(1H1\)')
C
C      CHECK FOR PREVIOUS BATCH OPTIMIZATION (ONLY ONE ALLOWED)
C      FILE BX.XXX ACTS AS A FLAG; IF IT EXISTS, NO OTHER OPTIMIZATION
C      IS PERMITTED.  THIS FILE IS CREATED IN OPTIM.
C
       INQUIRE(FILE='BZ.XXX',EXIST=XXX)
       IF(XXX) THEN
          OPEN(88,FILE='BZ.XXX')
          CLOSE (88,STATUS='DELETE')
          OPEN  (15,FILE='OPTIMIZE.BIN')
          CLOSE (15,STATUS='DELETE')
          GO TO 777
       ENDIF
```

```
C
      CALL OPTIM(DUMMY)
      INQUIRE (FILE='OPTIMIZE.BIN',EXIST=LOG)
      IF (LOG.EQV..TRUE.) GOTO 900
 777  IF (NEXTFL.NE.BLANK1) GO TO 800
      CLOSE (1,STATUS='DELETE')
      GO TO 900
 800  IF (NEXTFL.EQ.END) THEN
         CLOSE(1,STATUS='DELETE')
         GO TO 900
       ELSE
         WRITE (1) NEXTFL
         CLOSE (1,STATUS='KEEP')
      ENDIF
 900  CONTINUE
      CLOSE(55,STATUS='DELETE')
      END
```

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM TO ESTIMATE COSTS OF PIPES
C     REVISION DATE 83-05-03
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE PIPES
     1      (DCDIA, PIPLAB, PIPMAT)
C
      COMMON/DUM/VERGPM(5), VERHD(5), VTOTLC(5), VTOTMC(5), VERTHP(8),
     &    AGITHP(10), AGITLC(10), AGITMC(10), DIATNK(10), HTTNK(10),
     &    TNKLC(10), TNKMC(10), VOLTNK(10), GPMPMP(11), PMPHP(11),
     &    TPMPLC(11), TPMPMC(11), TPH3, HPVAC, THKHT, CMECLC, CMECMC, CON3LG,
     &    CON8LG, DIASIL, DISPHD, FILTAR, HPMILL, HPSLAK, HTELE2, HTSILS,
     &    RAKEHP, REHTLC, REHTMC, SOBLLC, SOBLMC, SOBM, TFILTL, TFILTM, THIKAR,
     &    THKDIA, TPHSLK, TOTTKL, TOTTKM, VOLSIL, SOBL, NOTNK(10), NOMOT(45),
     &    NOPMOT(45), NSPMOT(45), NGR18, NOFILT, NOPREP, NOSILO, NOTHIK,
     &    NSFILT
      COMMON/DUMM/PMPFDN(11), TNKFDN(10), SPGR(10), BLDLAB(9), BLDMAT(9),
     1 EQULAB(9), EQUMAT(9), FDNLAB(9), FDNMAT(9), STRLAB(9), STRMAT(9),
     2 FBLDSF, FILFDL, FILFDM, HPFDRV, HPFPMP, VSCTNK, VERFDN, ELE2L, ELE2M,
     3 CSLAKL, CSLAKM, GPMILL, PCONL1, PCONL2, PCONL4, PCONM1, PCONM2, PCONM4,
     4 PEQLC1, PEQLC2, PEQMC1, PEQMC2, PSTLL1, PSTLL2, PSTLL3, PSTLL4, PSTLM1,
     5 PSTLM2, PSTLM3, PSTLM4, TPSLAK, TSLKHP, CMILLL, CMILLM, GPMTRN, NSUMPR
      COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
     1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
     2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
     3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
     4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
     5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
     6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
      COMMON/CSTNDX/ICEPYR, CEPNDX, INVYR, IREVYR, RATLAB, RATMAT, ARM, ARL
      COMMON/FLOWS/GPM, MFLOW, WFLOW, TOTFLO
      REAL MFLOW(20, 12), WFLOW(20, 12), TOTFLO(20), GPM(20)
      COMMON/IND/ENGIN, ARCTEC, FLDEXP, FEES, CONT, START, CONINT, FILMAT(9)
      COMMON/PND1/PDEPTH, PMXEXC, DISTPD, ILINER, CLAYIN, CLAYUC, HYPLMC,
     1            HYPLLC, PSAMAX, ACREC, APRAIN, SAPILE
      COMMON/SCRUB/ISCRUB, IFOX, IADD, VLG, VTR, IFAN
      COMMON/SRHT/TSTEAM, HVS, VRH, WSTEAM, ARH, DPRH, URH, Q, NT, NB, PL, OD,
     &ARINCL, ARCORT, NINCLT, NCORTT, NINCLB, NCORTB, TEMPS
      DIMENSION ARH(6), DPRH(6), URH(6), NT(6), NB(6), OD(6), ARINCL(6),
     &ARCORT(6), NINCLT(6), NCORTT(6), NINCLB(6), NCORTB(6)
      COMMON/TRAN/NOTRAN, NOREDN, NTOTAL, PCNTRN, NORAW, NOUTIL, NOUNIT,
     1            NOPROD, NOSHIP, NSPREP, NSCOMP, NOCOMP, NOBYDU
      COMMON/WGCOMP/YWCO2, YWHCL, YWSO2, YWO2, YWN2, YWH2O, WVAPOR, NTRAIN
      REAL NTRAIN
C
      DIMENSION PIPMAT(9), PIPLAB(9), PIPEX(80), PIPEY(80)
      DIMENSION PIPIN1(80), PIPIN2(80)
      DATA PIPIN1/7489.36, 320.104, 4552.33, 689.75, 3261.88, 3380.67,
```

```
     &1826.82,87.647,2357.58,613.53,26670.4,17591.7,3364.15,971.743,
     &21926.26,14483.4,3364.15,971.74,1183.11,589.40,4145.32,1357.90,
     &436.95,969.2,1333.54,828.2,283.8,63.513,2389.58,11615.0,
     &4252.19,11754.5,700.38,666.88,1.3659,2.807,3.88,1.664,
     &16240.,3734.5,3741.6,584.32,1486.05,1272.79,3741.6,584.32,
     &1486.05,1272.79,3.88,1.66,150.34,54.62,1075.64,360.75,
     &713.25,223.564,3.88,1.66,3865.73,1476.03,3.88,1.66,
     &44.36,224.83,1771.26,1139.47,1.3655,2.8073,2084.93,1281.68,
     &2522.47,3674.84,43.88,22.445,150.34,54.621,1075.64,
     &360.75,713.247,215.94/
      DATA PIPIN2/.257947,.4528,.469269,.280447,.416544,.399396,
     &.353308,.396298,.177282,.19267,.43751,.262947,.272633,.188209,
     &.361478,.286086,.272633,.188209,.443309,.245169,.389936,.303329,
     &.33493,.3292,.414571,.255696,.41857,.2708,.378423,.308282,
     &.235404,.148409,.478149,.40712,.583853,.376378,.478149,.40712,
     &.583853,.376378,.378423,.308282,.541782,.406928,.527415,.411663,
     &.48411,.348086,.378423,.308282,.355653,.253727,.378423,.308282,
     &.4985,.2694,.412268,.302154,.41857,.2708,.486484,.270809,
     &.388316,.340924,.481095,.339238,.541782,.406928,.527415,
     &.41163,.48411,.348086/
C
CCC   GPM(12)=COOLING ISCRUB LT.3
CCC            VENTURI ISCRUB GE.3
CC    PIPING
C
      PI=3.1415926
      SPDOX = 0.0
      SPDOL = 0.0
      NFILT=0
      DO 5 I =1,80
      PIPEX(I) = 0.0
      PIPEY(I) = 0.0
    5 CONTINUE
      NTPREP=NOPREP+NSPREP
      SUM=(GPM(6)+WVAPOR+NTRAIN)/NOTRAN
      IF(ISCRUB.NE.3.AND.ISCRUB.NE.5) SUM=0.0
      IF(IFOX.EQ.3) SUM=0.0
      PIPEX(1)=GPM(8)
      PIPEX(2)=GPM(8)/NOPREP
      PIPEX(3)=GPM(8)
      PIPEX(4)=PIPEX(2)
      PIPEX(5)=GPM(9)
      PIPEX(6)=GPM(9)/NOTRAN
      PIPEX(7)=PIPEX(6)
      PIPEX(8)=GPM(9)
      PIPEX(9)=PIPEX(6)
      PIPEX(10)=PIPEX(6)
      PIPEX(11)=GPM(9)
      PIPEX(12)=GPM(9)
      PIPEX(13)=GPM(9)/NOPREP
      PIPEX(14)=GPM(9)/NOPREP
```

```
      PIPEX(15)=GPM(9)
      PIPEX(16)=GPM(9)
      PIPEX(17)=GPM(9)/NOPREP
      PIPEX(18)=GPM(9)/NOPREP
      PIPEX(19)=GPM(9)/NOPREP
      PIPEX(20)=GPM(9)/NOPREP
      PIPEX(21)=GPM(12)/NOTRAN
      PIPEX(22)=PIPEX(21)
      PIPEX(23)=GPM(10)/NOTRAN
      PIPEX(24)=PIPEX(23)
      PIPEX(25)=GPM(6)/NOTRAN
      PIPEX(26)=PIPEX(25)
      PIPEX(27)=GPM(11)/NOTRAN
      PIPEX(28)=PIPEX(27)
      RSTEAM = WSTEAM/1000.
      PIPEX(29)=RSTEAM
      PIPEX(30)=RSTEAM/NOTRAN
      PIPEX(31)=RSTEAM
      PIPEX(32)=PIPEX(30)
      PIPEX(33)=GPM(15)
      PIPEX(34)=GPM(15)
      PIPEX(35)=GPM(15)
      PIPEX(36)=GPM(15)
      IF(ISLUDG.LT.4) GO TO 1000
      PIPEX(35)=GPM(17)*2/NOFILT
      PIPEX(36)=PIPEX(35)
 1000 CONTINUE
      IF(ISLUDG.EQ.2) GO TO 1020
      DO 1010 KX=37,44
      PIPEX(KX)=GPM(6)
 1010 CONTINUE
      GO TO 1040
 1020 CONTINUE
      DO 1030 KX=37,44
      PIPEX(KX)=GPM(13)
 1030 CONTINUE
 1040 CONTINUE
      DO 1050 KX=45,52
      PIPEX(KX)=GPM(6)
 1050 CONTINUE
      DO 1060 KX=53,66
      PIPEX(KX)=0.0
 1060 CONTINUE
      IF (NOTHIK.EQ.0.0) GO TO 1090
      DO 1070 KX=53,58
      PIPEX(KX)=GPM(6)/NOTHIK
 1070 CONTINUE
      DO 1080 KX=59,62
      PIPEX(KX)=GPM(13)/NOTHIK
 1080 CONTINUE
      PIPEX(63)=GPM(14)
      PIPEX(64)=GPM(14)
```

```
      PIPEX(65)=GPM(14)*2/NOTHIK
      PIPEX(66)=PIPEX(65)
 1090 CONTINUE
      PIPEX(67)=GPM(7)
      PIPEX(68)=GPM(7)
      PIPEX(69)=GPM(13)
      PIPEX(70)=GPM(13)
      PIPEX(71)=GPM(7)
      PIPEX(72)=GPM(7)
      PIPEX(73)=SUM
      PIPEX(74)=PIPEX(73)
      DO 1100 KX=75,80
      PIPEX(KX)=GPM(6)/NOTRAN
 1100 CONTINUE
      DO 1110 KP=1,80
      PIPEY(KP)=PIPIN1(KP)*PIPEX(KP)**PIPIN2(KP)
 1110 CONTINUE
C              DRAIN    WATER    SERAIR     PIPE CONTINGENCY AS NOTED
      PIPERM=.08    +    .02    + 0.0    + 0.0
      PIPEFP=.02    +    .08    + 0.03   + 1.0
      PIPFAN=.00    +    0.0    + 0.0    + 0.0
      PIPESB=0.1    +    0.1    + .03    + 1.0
      PIPEOX=0.01   +    .02    + 0.01   + 1.0
      PIPERH=0.01   +    0.0    + 0.02   + 1.0
      PIPESS=0.075  +    0.05   + 0.03   + 1.0
C     PREP. AREA PIPING
      PPP8M=PIPEY(1)+NTPREP*PIPEY(2)
      PPP8L=PIPEY(3)+NTPREP*PIPEY(4)
      PPP9M=PIPEY(5)+NTOTAL*PIPEY(6)+NSPMOT(5)*PIPEY(7)
      PPP9L=PIPEY(8)+NTOTAL*PIPEY(9)+NSPMOT(5)*PIPEY(10)
      IF(GPM(9).GT.20.) GO TO 1120
      PHEADM=PIPEY(11)
      PHEADL=PIPEY(12)
      PPUMPM=PIPEY(13)
      PPUMPL=PIPEY(14)
      GO TO 1140
 1120 PHEADM=PIPEY(15)
      PHEADL=PIPEY(16)
      IF(PIPEX(17).GT.20.) GO TO 1130
      PPUMPM=PIPEY(17)
      PPUMPL=PIPEY(18)
      GO TO 1140
 1130 PPUMPM=PIPEY(19)
      PPUMPL=PIPEY(20)
 1140 CONTINUE
      NTRAWP=NTPREP
      IF (NSPREP.EQ.0) NTRAWP=NTRAWP+1
      PPBMM=PHEADM+NTRAWP*PPUMPM
      PPBML=PHEADL+NTRAWP*PPUMPL
      PIPMAT(1)=(PPP8M+PPP9M+PPBMM)*PIPERM*1.20*RATMAT
      PIPMAT(2)=(PPP8M+PPP9M+PPBMM)*PIPEFP*1.20*RATMAT
      PIPLAB(1)=(PPP8L+PPP9L+PPBML)*PIPERM*1.20*RATLAB
```

```
      PIPLAB(2)=(PPP8L+PPP9L+PPBML)*PIPEFP*1.20*RATLAB
C     SCRUBBER AREA PIPING
      SPP12M=NTOTAL*PIPEY(21)
      SPP12L=NTOTAL*PIPEY(22)
      SPP10M=NTOTAL*PIPEY(23)
      SPP10L=NTOTAL*PIPEY(24)
      SPP6M=NTOTAL*PIPEY(25)
      SPP6L=NTOTAL*PIPEY(26)
      SPP11M=NTOTAL*PIPEY(27)
      SPP11L=NTOTAL*PIPEY(28)
      SPEVTM=NTOTAL*PIPEY(73)
      SPEVTL=NTOTAL*PIPEY(74)
      SPP13M=PIPEY(29)+NTOTAL*PIPEY(30)
      SPP13L=PIPEY(31)+NTOTAL*PIPEY(32)
      DCDIA=SQRT((GPM(11)+GPM(12))*4./(450.*PI*NOTRAN))
      IF(ISCRUB.LT.3) GO TO 1150
      DCDIAA=SQRT((GPM(11)*4.0)/(450.0*PI*NOTRAN))
      DCDIAV=SQRT((GPM(12)*4.0)/(450.0*PI*NOTRAN))
      DCDIA=DCDIAA+DCDIAV
 1150 CONTINUE
      IF(IFOX.NE.2) GO TO 1151
      OVFOX = SQRT(((GPM(11)+GPM(12))*4.0)/(450.0*PI*NOTRAN))
      OXDUCT = PI*OVFOX*10*10.2*1.10
      OXSQFT = PI*OVFOX*10*1.10
      SPDOX = (OXDUCT*.56+OXSQFT*4.7)*NTOTAL
      SPDOL = (OXDUCT*1.78+OXSQFT*4.2)*NTOTAL
 1151 CONTINUE
      DCWT=PI*DCDIA*10*10.2*1.10
      DCSQFT=PI*DCDIA*10*1.10
      SPDCM=(DCWT*.56+DCSQFT*4.7)*NTOTAL
      SPDCL=(DCWT*1.78+DCSQFT*4.2)*NTOTAL
      SPSUBM=SPP12M+SPP10M+SPP6M+SPP11M+SPEVTM
      SPSUBL=SPP12L+SPP10L+SPP6L+SPP11L+SPEVTL
      SBOXDM=0.0
      SBOXDL=0.0
      IF(IFOX.NE.3) GO TO 1160
      SBOXDM=(PIPEY(75)+PIPEY(77))*NOTRAN+PIPEY(79)*NSPMOT(9)
      SBOXDL=(PIPEY(76)+PIPEY(78))*NOTRAN+PIPEY(80)*NSPMOT(9)
 1160 CONTINUE
      PIPMAT(3)=0.0
      PIPMAT(4)=(SPSUBM*PIPESB+SPDCM)*1.20*RATMAT
      PIPMAT(5)=(SBOXDM+SPDOX)*PIPEOX*1.20*RATMAT
      PIPMAT(6)=SPP13M*PIPERH*1.20*RATMAT
      PIPLAB(3)=0.0
      PIPLAB(4)=(SPSUBL*PIPESB+SPDCL)*1.20*RATLAB
      PIPLAB(5)=(SBOXDL+SPDOL)*PIPEOX*1.20*RATLAB
      PIPLAB(6)=SPP13L*PIPERH*1.20*RATLAB
C     SOLIDS DISPOSAL AREA
C
C     PONDING
      IF(ISLUDG.GT.2) GO TO 1200
      DPP7M=PIPEY(33)+DISTPD*PIPEY(35)
```

```
      DPP7L=PIPEY(34)+DISTPD*PIPEY(36)
C     POND OPTION
      DPP6M1=PIPEY(37)*2.*DISTPD
      DPP6L1=PIPEY(38)*2.*DISTPD
      IF(GPM(6).GT.255.) GO TO 1170
      DPP6M2=PIPEY(39)
      DPP6L2=PIPEY(40)
      GO TO 1190
 1170 IF(GPM(6).GT.20000.) GO TO 1180
      DPP6M2=PIPEY(41)
      DPP6L2=PIPEY(42)
      GO TO 1190
 1180 DPP6M2=PIPEY(43)
      DPP6M2=PIPEY(44)
 1190 CONTINUE
      IF(ISLUDG.GT.1) GO TO 1200
      PIPMAT(7)=(DPP7M+DPP6M1+DPP6M2)*PIPESS*1.20*RATMAT
      PIPLAB(7)=(DPP7L+DPP6L1+DPP6L2)*PIPESS*1.20*RATLAB
      GO TO 1330
 1200 CONTINUE
      IF (NOTHIK.GT.1) GO TO 1210
      DPP6M2=0.0
      DPP6L2=0.0
 1210 CONTINUE
      IF(GPM(6).GT.20000.) GO TO 1220
      DPPBM=PIPEY(45)+1000.*PIPEY(49)
      DPPBL=PIPEY(46)+1000.0*PIPEY(50)
      GO TO 1230
 1220 CONTINUE
      DPPBM=PIPEY(47)+1000.*PIPEY(49)
      DPPBL=PIPEY(48)+1000.*PIPEY(50)
 1230 CONTINUE
      IF(NOTHIK.GT.1) GO TO 1240
      DPPCM=0
      DPPCL=0
      GO TO 1280
 1240 CONTINUE
      NTNKC=NOTNK(5)
      NDEH2O=NOTHIK
      NMOTC=NSPMOT(18)
C     NFILT=0
 1250 CONTINUE
      IF(NFILT.EQ.0)GPMC=PIPEX(53)
      IF(GPMC.GT.70.) GO TO 1260
      DPPCL=NTNKC*PIPEY(52)+2770.*NDEH2O+PIPEY(56)*
     1NMOTC+200*PIPEY(58)*NDEH2O
 1260 CONTINUE
      IF(GPMC.GT.600.) GO TO 1270
      DPPCM=NTNKC*PIPEY(51)+(8960.+6.5*GPMC)*NDEH2O+
     &6350.*NMOTC+256.*PIPEY(57)*NDEH2O
      IF(GPMC.LE.70.) GO TO 1280
 1270 CONTINUE
```

```
      DPPCL=NTNKC*PIPEY(52)+PIPEY(54)*NDEH2O+PIPEY(56)
     &*NMOTC+200.*PIPEY(58)*NDEH2O
      IF(GPMC.LE.600.) GO TO 1280
      DPPCM=NTNKC*PIPEY(51)+PIPEY(53)*NDEH2O+PIPEY(55)*
     1NMOTC+200.*PIPEY(57)*NDEH2O
C     IF(NFILT.EQ.1) GO TO 1320
 1280 CONTINUE
      IF(NFILT.EQ.1) GO TO 1320
      DPPC1M=DPPCM
      DPPC1L=DPPCL
      DPPDGM=PIPEY(59)*NOTHIK+NOTHIK*3.5*THKDIA*
     1PIPEY(61)+PIPEY(63)*NOTHIK+PIPEY(65)*NOTNK(4)+1100.*PIPEY(67)
      DPPDGL=PIPEY(60)*NOTHIK+NOTHIK*3.5*THKDIA*
     1PIPEY(62)+PIPEY(64)*NOTHIK+PIPEY(66)*NOTNK(4)+1100.*PIPEY(68)
      DPPJM=PIPEY(71)
      DPPJL=PIPEY(72)
      IF(ISLUDG.EQ.3) GO TO 1290
      IF(ISLUDG.GE.4) GO TO 1300
      PIPMAT(7)=(((DPP7M+DPP6M1+DPP6M2+DPPBM+DPPC1M+
     1DPPDGM+DPPJM)*PIPESS)+DRAINM)*1.20*RATMAT
      PIPLAB(7)=(((DPP7L+DPP6L1+DPP6L2+DPPBL+DPPC1L+DPPDGL+
     1DPPJL)*PIPESS)+DRAINL)*1.20*RATLAB
      GO TO 1330
 1290 CONTINUE
      DPPIM=PIPEY(69)
      DPPIL=PIPEY(70)
      IF(ISLUDG.GE.4) GO TO 1300
      PIPMAT(7)=(((DPPBM+DPPC1M+DPPDGM+DPPJM+DPPIM)
     &*PIPESS)+DRAINM)*1.20*RATMAT
      PIPLAB(7)=(((DPPBL+DPPC1L+DPPDGL+DPPJL+DPPIL)*PIPESS)+DRAINL)*
     11.20*RATLAB
      GO TO 1330
 1300 CONTINUE
      IF(NOTHIK.EQ.1) GO TO 1310
      NFILT=1
      GPMC=GPM(13)/NOFILT
      NTNKC=NOTNK(6)
      NDEH2O=NOFILT
      NMOTC=NSPMOT(20)
      GO TO 1250
 1310 CONTINUE
      DPPCM=0
      DPPCL=0
 1320 CONTINUE
      DPPHFM=PIPEY(35)*NOTNK(7)+PIPEY(65)*
     1(NOFILT+NSFILT+NOTNK(7))
      DPPHFL=PIPEY(36)*NOTNK(7)+PIPEY(66)*(NOFILT+NSFILT+NOTNK(7))
      PIPMAT(7)=(((DPPBM+DPPC1M+DPPDGM+DPPJM+DPPCM+
     1 DPPHFM)*PIPESS)+DRAINM)*1.20*RATMAT
      PIPLAB(7)=(((DPPBL+DPPC1L+DPPDGL+DPPJL+DPPCL+DPPHFL)
     1*PIPESS)+DRAINL)*1.20*RATLAB
 1330 CONTINUE
```

```
RETURN
END
```

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C      SUBPROGRAM DUCWRK
C      REVISION DATE 83-08-30
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       SUBROUTINE DUCWRK
      1       (DCDIA, EXPLG, DUCLAB, DUCMAT, DUCTWD, HORNDL,
      2       PLENLG, SCRBLG, STKDIA)
C
       COMMON/DUM/VERGPM(5), VERHD(5), VTOTLC(5), VTOTMC(5), VERTHP(8),
      &    AGITHP(10), AGITLC(10), AGITMC(10), DIATNK(10), HTTNK(10),
      &    TNKLC(10), TNKMC(10), VOLTNK(10), GPMPMP(11), PMPHP(11),
      &    TPMPLC(11), TPMPMC(11), TPH3, HPVAC, THKHT, CMECLC, CMECMC, CON3LG,
      &    CON8LG, DIASIL, DISPHD, FILTAR, HPMILL, HPSLAK, HTELE2, HTSILS,
      &    RAKEHP, REHTLC, REHTMC, SOBLLC, SOBLMC, SOBM, TFILTL, TFILTM, THIKAR,
      &    THKDIA, TPHSLK, TOTTKL, TOTTKM, VOLSIL, SOBL, NOTNK(10), NOMOT(45),
      &    NOPMOT(45), NSPMOT(45), NGR18, NOFILT, NOPREP, NOSILO, NOTHIK,
      &    NSFILT
       COMMON/DUMM/PMPFDN(11), TNKFDN(10), SPGR(10), BLDLAB(9), BLDMAT(9),
      1 EQULAB(9), EQUMAT(9), FDNLAB(9), FDNMAT(9), STRLAB(9), STRMAT(9),
      2 FBLDSF, FILFDL, FILFDM, HPFDRV, HPFPMP, VSCTNK, VERFDN, ELE2L, ELE2M,
      3 CSLAKL, CSLAKM, GPMILL, PCONL1, PCONL2, PCONL4, PCONM1, PCONM2, PCONM4,
      4 PEQLC1, PEQLC2, PEQMC1, PEQMC2, PSTLL1, PSTLL2, PSTLL3, PSTLL4, PSTLM1,
      5 PSTLM2, PSTLM3, PSTLM4, TPSLAK, TSLKHP, CMILLL, CMILLM, GPMTRN, NSUMPR
       COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
      1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
      2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
      3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
      4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
      5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
      6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
       COMMON/BOILER/MW, BHR, EXSAIR, RMW, PREC, SPASH, WPRITE, KCLEAN, SMRW, SMCL
       REAL MW
       COMMON/BYPAS2/BYLBH(6), BYLBMH(6), BSCFM, BACFM, PASSBY, XPASSD, IBY
       COMMON/CSTNDX/ICEPYR, CEPNDX, INVYR, IREVYR, RATLAB, RATMAT, ARM, ARL
       COMMON/FLYASH/IASH, ASHUPS, ASHSCR, ASHPRU, ASHPRS, ASHLBU, ASHLBS
       COMMON/HGAS/ACFM, THG, YCO2, YHCL, YSO2, YO2, YN2, YH2O, ASH, WCL, WASH, IESP
       COMMON/IND/ENGIN, ARCTEC, FLDEXP, FEES, CONT, START, CONINT, FILMAT(9)
       COMMON/SCRUB/ISCRUB, IFOX, IADD, VLG, VTR, IFAN
       COMMON/SCRUBR/LG, V, PPMOUT, SO2R, ELIMIT, PR, PD, WPS, OX, PH, NSTAGE,
      &       NGRID, HS, ACS, ACSTRN
       REAL LG
       COMMON/SKGAS/SSCFM, SACFM, TSK, SDUST, SWDUST, SMPCT, SLBMH, SLBH, TGAS
       REAL SMPCT(6), SLBMH(6), SLBH(6)
       COMMON/SRHT/TSTEAM, HVS, VRH, WSTEAM, ARH, DPRH, URH, Q, NT, NB, PL, OD,
      &ARINCL, ARCORT, NINCLT, NCORTT, NINCLB, NCORTB, TEMPS
       COMMON/TRAN/NOTRAN, NOREDN, NTOTAL, PCNTRN, NORAW, NOUTIL, NOUNIT,
      1            NOPROD, NOSHIP, NSPREP, NSCOMP, NOCOMP, NOBYDU
       COMMON/WGAS/WACFM, WSCFM, TWG, ASHOUT, WASHOT, WMPCT, WLBMH, WLBH
```

```
      REAL WMPCT(6),WLBMH(6),WLBH(6)
      DIMENSION ARH(6),DPRH(6),URH(6),NT(6),NB(6),OD(6),ARINCL(6),
     &ARCORT(6),NINCLT(6),NCORTT(6),NINCLB(6),NCORTB(6)
C
      DIMENSION DUCLAB(9),DUCMAT(9)
C
C     DUCTWORK
      DAMER2 = 0.0
      DUCTOX = 0.0
      OUTL = 0.0
      OACFM = SACFM - BACFM
      PI=3.1415926
      SCRBWD=SQRT(ACSTRN/2.)
      SCRBLG=2.*SCRBWD
      DUCTWD=SQRT(ACFM/(NOTRAN*3000.))
      DUCOUT=SQRT(OACFM/(NOTRAN*3000.))
      DUCPLG=SQRT(SACFM/(NOTRAN*3000.))
      ACGAS = ACFM / NOTRAN
      ASGAS = OACFM / NOTRAN
      DAMA = 263.778
      DAMB = 0.417109
      IF(ASHUPS.GE.90.) GO TO 100
      DAMA = 45.7405
      DAMB = 0.589061
  100 DAMER1 = 1.15* NTOTAL * DAMA * ACGAS**DAMB
      IF ( TEMPS. GT. 150) GO TO 200
      DAMA = 45.7405
      DAMB = 0.589061
  200 DAMER3 = 1.15* NTOTAL * DAMA * ASGAS**DAMB
      REHTHT = 10.
      REHTV=VRH
      REHTWD=SQRT(WACFM/(NOTRAN*REHTV*60))
      STKDIA=2.*SQRT(SACFM*4./(PI*3600.))
      PLENLG=STKDIA+2.*DUCTWD+5*NTOTAL+(NTOTAL-2)*SCRBLG
      VERNDL=HTTNK(2)+6.103+(.513*SCRBLG-.204*DCDIA)
      IF(ISCRUB.GE.3)VERNDL=VERNDL+33
      IF(ISCRUB.LT.3)GO TO 1000
      EXPLG=(SCRBLG-8-DUCTWD)/1.1547-10
      IF (EXPLG.GE.10)GO TO 1010
      EXPLG=0
 1000 CONTINUE
      EXPLG=(SCRBLG-4.-DUCTWD)/1.1547-10.
      IF(EXPLG.GE.10.)GO TO 1010
      EXPLG=0
 1010 SAIDE=EXPLG*(2.*DUCTWD+EXPLG/.866)+2.*DUCTWD*EXPLG/.866
      HORNDL=DIATNK(2)-.5*SCRBWD-.5*DCDIA+DIATNK(8)-.224*DUCTWD
     &-.9659*(EXPLG+10.)+14.
C
C     ADD  + DIATNK(9)     FAS  6-22-80
C
      IF(IFAN.EQ.1) HORNDL = HORNDL - 14.
      IF(HORNDL.GE.0.)GO TO 1020
```

```
      HORNDL=0
 1020 STORHT=(SCRBLG-REHTWD)/2.
      IF(REHTWD.GT.(1.5*SCRBWD))STORHT=(REHTWD-SCRBWD)/2.
      SASTOR=(SCRBLG+REHTWD)*SQRT(STORHT**2+((SCRBWD-REHTWD)/2.)**2)
     &       +(SCRBWD+REHTWD)*SQRT(STORHT**2+((SCRBLG-REHTWD)/2.)**2)
      RTODHT=(REHTWD-DUCTWD)/2.
      SARTOD=2.828*RTODHT*(REHTWD+DUCTWD)
      OUTDLG=69.+.5*SCRBWD+.789*SCRBLG+1.948*DUCPLG+
     &REHTHT+.204*DCDIA+.5*STKDIA
      IF(IFAN.EQ.1) OUTDLG = OUTDLG + 21.
      IF(HTTNK(2).GE.30.0.OR.HTTNK(10).GE.30.0)GO TO 1030
      NSCRUB=2
      IF (ISCRUB.EQ.4.OR.ISCRUB.EQ.6.OR.IFOX.EQ.1) NSCRUB=8
      OUTDLG=OUTDLG+HTTNK(NSCRUB)-30.
 1030 TRNDUC=VERNDL+HORNDL
      IF(IFOX.EQ.2) DUCTOX = DIATNK(10)*4.*DUCTWD*NTOTAL*1.15
      IF(TEMPS.GE.150) OUTL = OUTDLG * NTOTAL
      DUCTLG=PLENLG+NTOTAL*TRNDUC
      SACORT=1.15*(4*DUCTWD*DUCTLG+4*DUCOUT*OUTL+NTOTAL*(SAIDE+SARTOD))
      IF(TEMPS.LT.150) SASTOR = SASTOR + (OUTL*DUCOUT*4./NTOTAL)
      SASSTL=1.15*NTOTAL*(SASTOR+5*REHTWD*REHTHT)
      EXPJON=12
      EXPJOT=16
      IF(ISCRUB.GE.3)EXPJON=16
      EXPJNT=EXPJON*NTOTAL*DUCTWD*1.15
      EXPJN2=EXPJOT*NTOTAL*DUCOUT*1.15
      EXPJNT = EXPJNT + EXPJN2
C
      IF(KEPASS.NE.1.AND.KPASO2.NE.1) GO TO 1040
      EBACFM=BACFM
      IF(KEPASS.EQ.1) EBACFM=ACFM*EBYPAS/100.0
      NOBYDU = 2
      IF(EBACFM.LE.96000) NOBYDU = 1
      IF(NOBYDU.GT.NTOTAL) NOBYDU = NTOTAL
      EBACFM = EBACFM / NOBYDU
      BYDUWD = SQRT(EBACFM/(NOBYDU*3000))
      BYDUCT = DIATNK(2) + DIATNK(10) + 90.
      EXJONB = 8. * NOBYDU * BYDUWD * 1.15
      DAMC = 80.7489
      DAMD = 0.4424
      DAMER2 = 2 * 1.15* NOBYDU *  DAMC * EBACFM**DAMD
      EXPJNT = EXPJNT + EXJONB
      PLENBY = (5 * NOBYDU + BYDUWD + (.5 * SCRBLG)) * DUCTWD
      SACORT = SACORT + (1.15 *4 * (PLENBY + BYDUWD *BYDUCT * NOBYDU))
      ELJOIN = EXJONB*10.
      EMJOIN = EXJONB *85.
      EMDAMP = DAMER2
      ELDAMP = DAMER2*0.1
      EMCORT = (1.15*4*(PLENBY + BYDUWD*BYDUCT *NOBYDU))
      EMCORC = EMCORT * 11.42
      ELCORC = EMCORT * 19.62
      EMBPAS = (EMJOIN+EMDAMP+EMCORC) * RATMAT
```

```
      ELBPAS = (ELJOIN+ELDAMP+ELCORC) * RATLAB
C
 1040 CONTINUE
C     PRICES INCLUDE 2 INCH INSULATION
C     PRICES EXPANSION JOINTS 85/FT 304 INLET, 85/FT FABRIC OUTLET
      DAMPER = DAMER1 + DAMER2 + DAMER3
      DUCMAT(1)=0.
      DUCMAT(2)=0.
      DUCMAT(3)=(SACORT*11.42+SASSTL*29.92+EXPJNT*85.
     &           +DAMPER)*RATMAT
      DUCMAT(4)=0.
      DUCMAT(5)=DUCTOX*11.42*RATMAT
      DUCMAT(6)=0.
      DUCMAT(7)=0.
      DUCLAB(1)=0.
      DUCLAB(2)=0.
      DUCLAB(3)=((SACORT+SASSTL)*19.62+EXPJNT*10.+DAMPER*0.11)*RATLAB
 1099 FORMAT(//' GAS HANDLING DUCT MATERIAL COST: ',1PG12.3,
     &        /' GAS HANDLING DUCT LABOR COST: ',G12.3//)
      DUCLAB(4)=0.
      DUCLAB(5)=DUCTOX*19.62*RATLAB
      DUCLAB(6)=0.
      DUCLAB(7)=0.
      RETURN
      END
```

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM INSTRM
C     REVISION DATE 83-03-22
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE INSTRM
     1      (ENSLAB,ENSMAT)
C
      COMMON/DUM/VERGPM(5),VERHD(5),VTOTLC(5),VTOTMC(5),VERTHP(8),
     &   AGITHP(10),AGITLC(10),AGITMC(10),DIATNK(10),HTTNK(10),
     &   TNKLC(10),TNKMC(10),VOLTNK(10),GPMPMP(11),PMPHP(11),
     &   TPMPLC(11),TPMPMC(11),TPH3,HPVAC,THKHT,CMECLC,CMECMC,CON3LG,
     &   CON8LG,DIASIL,DISPHD,FILTAR,HPMILL,HPSLAK,HTELE2,HTSILS,
     &   RAKEHP,REHTLC,REHTMC,SOBLLC,SOBLMC,SOBM,TFILTL,TFILTM,THIKAR,
     &   THKDIA,TPHSLK,TOTTKL,TOTTKM,VOLSIL,SOBL,NOTNK(10),NOMOT(45),
     &   NOPMOT(45),NSPMOT(45),NGR18,NOFILT,NOPREP,NOSILO,NOTHIK,
     &   NSFILT
      COMMON/DUMM/PMPFDN(11),TNKFDN(10),SPGR(10),BLDLAB(9),BLDMAT(9),
     1 EQULAB(9),EQUMAT(9),FDNLAB(9),FDNMAT(9),STRLAB(9),STRMAT(9),
     2 FBLDSF,FILFDL,FILFDM,HPFDRV,HPFPMP,VSCTNK,VERFDN,ELE2L,ELE2M,
     3 CSLAKL,CSLAKM,GPMILL,PCONL1,PCONL2,PCONL4,PCONM1,PCONM2,PCONM4,
     4 PEQLC1,PEQLC2,PEQMC1,PEQMC2,PSTLL1,PSTLL2,PSTLL3,PSTLL4,PSTLM1,
     5 PSTLM2,PSTLM3,PSTLM4,TPSLAK,TSLKHP,CMILLL,CMILLM,GPMTRN,NSUMPR
      COMMON/BIG1/BAGDLP,BAGRAT,BAGCOST,BGLIFE,PSSO2X,EBYPAS,RS,PSC,
     1 CCSA,EFFPS,ESPDLP,RESIST,SCARAT,FANHP,FANLC,FANMC,DELTAP,FILTSA,
     2 PSF,FILRAT,H,TL,VD,VE,TGASID,SKLAND,DRAINM,DRAINL,CLAYSK,WPSO3,
     3 WPSO4,WPCO3,WPINS,ASHCLN,WINDEX,PDMAX,FLOOD,SRAIR,OXAIR,OXSCFM,
     4 PRES,SO2RT,SO2RV,SO2RST,SDFEE,VPD,VT,HVCLN,HPTONW,
     5 ISLUDG,IFIXS,JSR,IVPD,IT,LON,LGMAX,ISTAND,NSBR,
     6 KEPASS,KPASO2,ICLAR,JSSO,JDIS,JSTR,JGPM,IRH,NSBF
      COMMON/CSTNDX/ICEPYR,CEPNDX,INVYR,IREVYR,RATLAB,RATMAT,ARM,ARL
      COMMON/FLOWS/GPM,MFLOW,WFLOW,TOTFLO
      COMMON/FIXFDN/ALCONY,FDNICS,PUGFDN,NSILFX,NASHS
      COMMON/IND/ENGIN,ARCTEC,FLDEXP,FEES,CONT,START,CONINT,FILMAT(9)
      COMMON/RTIME/TR,ALCONT(9)
      COMMON/SCRUB/ISCRUB,IFOX,IADD,VLG,VTR,IFAN
      COMMON/SRHT/TSTEAM,HVS,VRH,WSTEAM,ARH,DPRH,URH,Q,NT,NB,PL,OD,
     &ARINCL,ARCORT,NINCLT,NCORTT,NINCLB,NCORTB,TEMPS
      COMMON/TRAN/NOTRAN,NOREDN,NTOTAL,PCNTRN,NORAW,NOUTIL,NOUNIT,
     1           NOPROD,NOSHIP,NSPREP,NSCOMP,NOCOMP,NOBYDU
      REAL MFLOW(20,12),WFLOW(20,12),TOTFLO(20),GPM(20)
      DIMENSION ARH(6),DPRH(6),URH(6),NT(6),NB(6),OD(6),ARINCL(6),
     &ARCORT(6),NINCLT(6),NCORTT(6),NINCLB(6),NCORTB(6)
C
      DIMENSION ENSMAT(9),ENSLAB(9)
C
C     SCRUBBER AREA
      A = VTR/4
      SVARM1=0.0
```

```
      SVARM4=0.0
      SVARM5=0.0
      SVARLV=0.0
      SFIXLV=0.0
      SINFOM=0.0
      SINFOL=0.0
      SINHAT=5520.*NTOTAL
      SINFAN=10375.*NTOTAL
      SIMFIX=99020.
      SILHAT=1025.*NTOTAL
      SILFAN=1665.*NTOTAL
      SILFIX=16000.
      IF(IRH.NE.0)SVARM1=NTOTAL*1874.0*(WSTEAM/NOTRAN)**0.127146
      IF(IESP.NE.1) GO TO 1010
      SIMFIX = SIMFIX + 2800.
      SILFIX = SILFIX + 320.
 1010 CONTINUE
      SFIXM=NTOTAL*SIMFIX+4000.
      B = GPM(11)
      IF(IFOX.EQ.1) B = ((A/(TR+A)) +1.0) * GPM(11)
      SVARM2=NTOTAL*255.098*(B/NOTRAN)**0.436137
      IF(ISCRUB.LT.3.AND.IFOX.EQ.0)GO TO 1020
      GPMI=GPM(12)
      SVARM4=NTOTAL*255.098*(GPMI/NOTRAN)**0.436137
      IF(IFOX.EQ.3)SVARM4=NTOTAL*255.098*(GPM(13)/NOTRAN)**0.436137
      SVARM5=NTOTAL*17888.5
      IF(ISCRUB.EQ.4.OR.ISCRUB.EQ.6.OR.IFOX.NE.2)SVARM5=NTOTAL*11278
 1020 CONTINUE
      GPM10B=GPM(10)/FLOAT(NOTRAN)
      IF(GPM10B.GT.800.) GO TO 1030
      SVARM3=NTOTAL*2800
      GO TO 1040
 1030 SVARM3=NTOTAL*39.079*GPM10B**0.610424
 1040 NWASH=0
      IF(NWASH.EQ.0) GO TO 1070
C     WASH TRAY OPTION
      IF(GPM10B.GT.380.) GO TO 1050
      WASHM=NTOTAL*9375.
      GO TO 1060
 1050 WASHM=NTOTAL*(340.868*GPM10B**0.385893+6783.)
 1060 WASHL=NTOTAL*1730.
      GO TO 1080
 1070 WASHM=0.0
      WASHL=0.0
 1080 SFIXL=NTOTAL*SILFIX+512.
      SVARL=NTOTAL*256.
      IF(IFOX.EQ.0.AND.ISCRUB.LT.3) GO TO 1110
      SINFOM=12770.
      SINFOL=1613.
      IF(IFOX.NE.3) GO TO 1090
      SINFOM=SINFOM/NTOTAL
      SINFOL=SINFOL/NTOTAL
```

```
 1090 IF(IFOX.NE.1.AND.ISCRUB.EQ.4.OR.ISCRUB.EQ.6) GO TO 1100
      SINFOM=11937.
      SINFOL=1421.
 1100 SINFOM=SINFOM*NTOTAL
      SINFOL=SINFOL*NTOTAL
 1110 CONTINUE
      IF(ISCRUB.LT.3)GO TO 1120
      SVARLV=NTOTAL*192.
      SFIXLV=4025*NTOTAL
      IF(ISCRUB.EQ.4.OR.ISCRUB.EQ.6)SFIXLV=2816*NTOTAL
      IF(ISCRUB.EQ.3.OR.ISCRUB.EQ.5)SFIXLV=SFIXLV+SINFOL
      IF(ISCRUB.EQ.3.OR.ISCRUB.EQ.5)SVARLV=SVARLV+ SINFOM
      SINFOL = 0.0
      SINFOM = 0.0
 1120 CONTINUE
C     INSTRUMENTS FOR BYPASS OPTION
      BYINSM = (25536. + 3059.) * NOBYDU
      BYINSL = (3328. + 640.) * NOBYDU
      SINSM=SFIXM+SVARM2+SVARM3+SVARM4+SVARM5+WASHM+BYINSM
      IF(IFOX.EQ.3)SINSM=SFIXM+SVARM2+SVARM3+SVARM5+WASHM+BYINSM
      SINSL=SFIXL+SFIXLV+SVARL+WASHL+BYINSL
C     PREP.AREA
      NTPREP=NOPREP+NSPREP
      PINSM=(NTPREP*30324.)+(NTOTAL*10041.)+4123.
      PINSL=(NTPREP*3392.)+(NTOTAL*1280.)+512.
      IF(IADD.EQ.0) GO TO 1130
      PINSM=PINSM + 3100.
      PINSL = PINSL + 721.
 1130 CONTINUE
C     SOLIDS DISPOSAL AREA
      DVARM1=0.0
      DVARM2=0.0
      DFIXM1=0.0
      DFIXM2=0.0
      DFIXL1=0.0
      DFIXL2=0.0
      DVARL=0.0
      IF(ISLUDG.GT.2) GO TO 1150
      IF(ISLUDG.EQ.1)GPMIN=GPM(6)
      IF(ISLUDG.EQ.2)GPMIN=GPM(13)
      DFIXM1=8140.
      IF(GPMIN.GT.800.) GO TO 1140
      DVARM1=665.
      DFIXL1=832.
      GO TO 1150
 1140 DVARM1=862.546*GPMIN**0.176427
 1150 CONTINUE
      IF(ISLUDG.EQ.1) GO TO 1190
      NFIXA=NOPMOT(23)+NOPMOT(24)+NOPMOT(25)+NOPMOT(26)
      NFIXB=NOMOT(10)+NOMOT(23)+NOMOT(24)+NOMOT(25)+NOMOT(26)
      DFIXM2=8047.+(NOTNK(4)+NOTNK(7))*1663.+6384.+NFIXA*
     14323.+NFIXB*146.
```

```
      DFIXL2=1024.+(NOTNK(4)+NOTNK(7))*384.+640.+NFIXA*
     1781.+NFIXB*32.
      GPMINS=GPM(6)
      DO 1180 I=23,26
      IF(NOPMOT(I).EQ.0) GO TO 1180
      IF(I.GE.25)GPMINS=GPM(13)
      IF(GPMINS.GT.400.) GO TO 1160
      DVARM2=446+DVARM2
      DVARL=83+DVARL
      GO TO 1180
 1160 CONTINUE
      IF(GPMINS.GT.2500.) GO TO 1170
      DVARM2=958.+DVARM2
      DVARL=175.+DVARL
      GO TO 1180
 1170 CONTINUE
      DVARM2=6.98368*GPMINS**0.5997 + DVARM2
      DVARL=9.42756*GPMINS**0.37354+DVARL
 1180 CONTINUE
 1190 CONTINUE
      DINSM=DFIXM1+DFIXM2+DVARM1+DVARM2
      DINSL=DFIXL1+DFIXL2+DVARL
C     MISCILLANEOUS INSTRUMENTATION PRORATED FOR AREAS 1 & 2
      MISCM=19847.9*NTOTAL**0.701754
      MISCL=6476.0*NTOTAL**0.951213
      STINSM=PINSM+SINSM+SINHAT+SINFAN
      IF(IFOX.EQ.3)STINSM=PINSM+SINSM+SINHAT+SINFAN+SVARM4
      STINSL=PINSL+SINSL+SILHAT+SILFAN
      NOITK1=NOTNK(3)+NOTNK(4)+NOTNK(5)+NOTNK(6)+NOTNK(7)
      PBFT=((NOITK1*3+NOTHIK)*3+(NOTNK(3)+NOTNK(5)+NOPMOT(24)
     &+NOTHIK+NOPMOT(25)+NOTNK(6)+NOPMOT(26))*6)/(3*12)+1
      NOWA=NOTNK(3)+2*NOTNK(5)+NOPMOT(23)+NOPMOT(24)+
     &NOTHIK+NOTNK(4)+NOPMOT(25)
      NOWB=2*NOTNK(6)+NOTNK(7)+NOPMOT(26)
      DMISCM=NOITK1*67.+PBFT*(1264.+232.)+(NOWA*1330.+
     &NOWB*532)*0.47+(NOWA*1064.+NOWB*266.)*0.34
      DMISCL=(NOWA*1280.+NOWB*512)*0.79+(NOWA*1024.+
     &NOWB*256.)*7.92
C     TOTAL AREA INSTRUMENTATION COSTS
      ENSMAT(1)=(0.1*PINSM/STINSM*MISCM)*RATMAT
      ENSLAB(1)=(0.1*PINSL/STINSL*MISCL)*RATLAB
      ENSMAT(2)=(PINSM+(0.9*PINSM/STINSM)*MISCM)*RATMAT
      ENSMAT(3)=(SINFAN+SINFAN/STINSM*MISCM)*RATMAT
      ENSMAT(4)=(SINSM+SINSM/STINSM*MISCM)*RATMAT
      ENSMAT(5)=(SINFOM+SINFOM/STINSM*MISCM)*RATMAT
      IF(IRH.NE.0)ENSMAT(6)=(SINHAT+SINHAT/STINSM*MISCM)*RATMAT
      ENSLAB(2)=(PINSL+(0.9*PINSL/STINSL)*MISCL)*RATLAB
      ENSLAB(3)=(SILFAN+SILFAN/STINSL*MISCL)*RATLAB
      ENSLAB(4)=(SINSL+SINSL/STINSL*MISCL)*RATLAB
      ENSLAB(5)=(SINFOL+SINFOL/STINSL*MISCL)*RATLAB
      IF(IRH.NE.0)ENSLAB(6)=(SILHAT+SILHAT/STINSL*MISCL)*RATLAB
      ENFIXM = 0.0
```

```
      ENFIXL = 0.0
      IF(IFIXS.LT.1) GO TO 2000
      NS = NASHS + NSILFX
      ENFIXM = (NS*3060.) + ((NS+1)*3046.)
      ENFIXL = (NS*1345.) + ((NS+1)*720.)
2000  CONTINUE
      ENSMAT(7)=(DINSM+DMISCM+ENFIXM)*RATMAT
      ENSLAB(7)=(DINSL+DMISCL+ENFIXL)*RATLAB
      RETURN
      END
```

```
$NOFLOATCALLS
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM ELECTR
C     REVISION DATE 83-09-30
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE ELECTR
     1       (REVHP,ELELAB,ELEMAT)
C
      COMMON/DUM/VERGPM(5),VERHD(5),VTOTLC(5),VTOTMC(5),VERTHP(8),
     &   AGITHP(10),AGITLC(10),AGITMC(10),DIATNK(10),HTTNK(10),
     &   TNKLC(10),TNKMC(10),VOLTNK(10),GPMPMP(11),PMPHP(11),
     &   TPMPLC(11),TPMPMC(11),TPH3,HPVAC,THKHT,CMECLC,CMECMC,CON3LG,
     &   CON8LG,DIASIL,DISPHD,FILTAR,HPMILL,HPSLAK,HTELE2,HTSILS,
     &   RAKEHP,REHTLC,REHTMC,SOBLLC,SOBLMC,SOBM,TFILTL,TFILTM,THIKAR,
     &   THKDIA,TPHSLK,TOTTKL,TOTTKM,VOLSIL,SOBL,NOTNK(10),NOMOT(45),
     &   NOPMOT(45),NSPMOT(45),NGR18,NOFILT,NOPREP,NOSILO,NOTHIK,
     &   NSFILT
      COMMON/DUMM/PMPFDN(11),TNKFDN(10),SPGR(10),BLDLAB(9),BLDMAT(9),
     1 EQULAB(9),EQUMAT(9),FDNLAB(9),FDNMAT(9),STRLAB(9),STRMAT(9),
     2 FBLDSF,FILFDL,FILFDM,HPFDRV,HPFPMP,VSCTNK,VERFDN,ELE2L,ELE2M,
     3 CSLAKL,CSLAKM,GPMILL,PCONL1,PCONL2,PCONL4,PCONM1,PCONM2,PCONM4,
     4 PEQLC1,PEQLC2,PEQMC1,PEQMC2,PSTLL1,PSTLL2,PSTLL3,PSTLL4,PSTLM1,
     5 PSTLM2,PSTLM3,PSTLM4,TPSLAK,TSLKHP,CMILLL,CMILLM,GPMTRN,NSUMPR
      COMMON/BIG1/BAGDLP,BAGRAT,BAGCOST,BGLIFE,PSSO2X,EBYPAS,RS,PSC,
     1 CCSA,EFFPS,ESPDLP,RESIST,SCARAT,FANHP,FANLC,FANMC,DELTAP,FILTSA,
     2 PSF,FILRAT,H,TL,VD,VE,TGASID,SKLAND,DRAINM,DRAINL,CLAYSK,WPSO3,
     3 WPSO4,WPCO3,WPINS,ASHCLN,WINDEX,PDMAX,FLOOD,SRAIR,OXAIR,OXSCFM,
     4 PRES,SO2RT,SO2RV,SO2RST,SDFEE,VPD,VT,HVCLN,HPTONW,
     5 ISLUDG,IFIXS,JSR,IVPD,IT,LON,LGMAX,ISTAND,NSBR,
     6 KEPASS,KPASO2,ICLAR,JSSO,JDIS,JSTR,JGPM,IRH,NSBF
      COMMON/ADDPRE/ADDSIL,DIAADD,HTSADD,IANGLE,ASILMC,ASILLC,PCOVMC,
     1PCOVLC,PCOVHP,ABINMC,ABINLC,AWFBMC,AWFBLC,ADUSTM,ADUSTL
      COMMON/CSTNDX/ICEPYR,CEPNDX,INVYR,IREVYR,RATLAB,RATMAT,ARM,ARL
      COMMON/FILCON/FCCTLC,FCCTMC,HPFCCM,LNHFCC,LNIFCC
      COMMON/FIXFDN/ALCONY,FDNICS,PUGFDN,NSILFX,NASHS
      COMMON/FIXCOS/FIXADD,DS,TFIXEQ,TFIXLB,HPFIX(8)
      COMMON/LS/IALK,ISR,SR,SRADD,WPMGO,WPI,WPM,LSADD,LIMEAD,MGOADD,WMG
      COMMON/PND1/PDEPTH,PMXEXC,DISTPD,ILINER,CLAYIN,CLAYUC,HYPLMC,
     1           HYPLLC,PSAMAX,ACREC,APRAIN,SAPILE
      COMMON/SCROXD/COMPHP,COMCFM,SPRDIA,COPRMC,COPRLC,SPARMC,SPARLC,FNO
      COMMON/SCRUB/ISCRUB,IFOX,IADD,VLG,VTR,IFAN
      COMMON/TRAN/NOTRAN,NOREDN,NTOTAL,PCNTRN,NORAW,NOUTIL,NOUNIT,
     1           NOPROD,NOSHIP,NSPREP,NSCOMP,NOCOMP,NOBYDU
      REAL LSADD,LIMEAD,MGOADD
C
      DIMENSION AMOTDT(8),ARHP(8),CIRBLC(8),CIRBMC(8),
     1          CONDLC(8),CONDMC(8),CTRLC(8),CTRMC(8),
     2          HP480(8),NLMOT(8),NMMOT(8),NOTMOT(8),NSMOT(8),
```

```
     3          OPHP(8),POWL(8),POWM(8),POWMLC(8),POWMMC(8),
     4          POWSLC(8),POWSMC(8),PSARHP(8),PSDIST(8),
     5          REMCTR(8),SPHP(8),STRM(8),
     6          TRFRLC(8),TRFRMC(8),WIRELC(8),WIREMC(8)
      DIMENSION ELELAB(9),ELEMAT(9)
      DIMENSION COST(7),RHP(10)
      DIMENSION EHP(45),INOMOT(45)
C
      DIMENSION ELECT(10,7,2)
      DATA ELECT/355.,370.,524.,2*762.,2*1672.,2*3785.,6375.,
     &          0.20,0.27,0.42,6*0.89,0.00205602,
     &          0.92,0.96,1.02,6*1.10,0.00514005,
     &          0.72,0.86,1.16,6*1.86,0.0076956,
     &          7*1.,2.,2*3.,
     &          190.,300.,510.,900.,2*1790.,3*70.7362,471.027,
     &          5*186.311,3*0.27048,2*204.728,
     &          10*0.0,
     &          9*0.0,1.00048,
     &          9*0.0,1.00048,
     &          9*0.0,0.948743,
     &          10*0.0,
     &          6*0.0,3*0.89287,0.639446,
     &          5*0.33909,3*1.80974,2*0.855260/
      DATA RHP/2.5,7.5,15.,35.,50.,90.,133.,233.,250.,1500./
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     ELECTRICAL  ARRAY ELECT  DEFINED  10,7,2
C                       DEFINATION
C     ELEMENT              HP
C        1                  2.5
C        2                  7.5
C        3                  15
C        4                  35
C        5                  50
C        6                  90
C        7                 133
C        8                 233
C        9                 250
C       10                 600
C * * * ** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C        1              STARTERS
C        2              POWER WIRE MATERIAL
C        3              POWER WIRE LABOR
C        4              CONDUIT MATERIAL
C        5              MOTOR SIZE  0-250 - S, 400-M, >400-L
C        6              MOTORS INVESTMENT
C        7              DRIVE INVESTMENT
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C    EQUATION  Y = A*X**B
C        1              A   NUBMER FOR EQUATION
C        2              B   POWER NUMBER FOR EQUATION
C * * * * * *  * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
C
C       MOTOR       DESCRIPTION      AREA
C         1         CONVEYORS          1
C         2         MILL/SLAKER        2
C         3         PRODUCT FEED       2
C         4         FANS               3
C         5         SLURRY FEED PUMP   2
C         6         ADD. PUMP          2  NOT USED NOW
C         7         COOL/VENTURI PUMP  4
C         8         SCRUB PUMP         4
C         9         OX BLEED PUMP      5
C        10         POND FEED PUMP     4
C        11         SLURRY FEED AGIT   2
C        12         OX WASTE AGIT      5
C        13        EFFLUENT TANK AGIT  4
C        14         VENTURI AGIT       4
C        15         RECIR. TANK AGIT   5
C        16         COMPRESSORS        5
C        17         POND FEED AGIT     4
C        18         MAKEUP WATER PUMP  4
C        19         POND SUPER PUMP    4
C        20         THICK OVERFLOW     7
C        21         FILTRATE PUMP      7
C        22         TOTAL FILTRATE     7
C        23         FEED TO THICK      7
C        24         THICK FEED PUMP    7
C        25         THICK UNDER PUMP   7
C        26         FILTER FEED        7
C        27         THICK FEED AGIT    7
C        28         THICK UNDER AGIT   7
C        29         RAKE               7
C        30         VACCUMM            7
C        31         DRIVE              7
C        32         PUMP               7
C        33         ADD. P.CONV        2
C        34         FILTER CONVEYOR    7
C        35         PNEUMATIC CONVEYOR FIXATION  7
C        36         LIME SCREW CONVEYOR          7
C        37         ASH SCREW CONVEYOR           7
C        38         ASH/LIME SCREW CONVEYOR      7
C        39         PUG MILL MOTOR               7
C        40         PUG MILL DUST COLLECTOR      7
C        41         DISCHARGE CONVEYOR           7
C        42         RADIAL STACKER               7
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      NTPREP=NOPREP+NSPREP
      IF(IALK.EQ.2) GO TO 1030
      IF(TPH3.GT.18.)GO TO 1000
      PSTRM1=0.
      PPOWM1=0.
```

```
      PPOWL1=0.
      P480A=0.
      PSMOT1=0.
      GO TO 1010
 1000 CONTINUE
      PSTRM1=4680.
      PPOWM1=9.63
      PPOWL1=74.81
      P480A=134.5
      PSMOT1=10.
 1010 PSTRM2=2366.+(NGR18+NSUMPR)*355.
      PPOWM2=5.03+(NGR18+NSUMPR)*.92
      PPOWL2=42.22+(NGR18+NSUMPR)*8.14
      P480B=78.5+NGR18*3.5+NSUMPR*5.
      PSMOT2=5.+NGR18+NSUMPR
      PSTRM6=NOPREP*2397.
      PPOWM6=NOPREP*3.83
      PPOWL6=NOPREP*26.14
      P480F=NOPREP*84.5
      PSMOT6=NOPREP*3.
      NOEMIL=NOPREP
      IF(NOPREP.LE.3)GO TO 1020
      NOPREP=3
 1020 PSTRM3=(NOPREP-1)*355.
      PPOWM3=(NOPREP-1)*.890
      PPOWL3=(NOPREP-1)*8.14
      P480C=(NOPREP-1)+(NOPREP-2)*6.5
      PSMOT3=NOPREP-1
      IF(NOEMIL.GT.3)NOPREP=NOEMIL
      PSTRM4=0.
      PPOWM4=0.
      PPOWL4=0.
      P480D=0.
      PSMOT4=0.
      EHP(1)=NGR18*2.5
      EHP(2)=HPMILL
      GO TO 1100
 1030 CONTINUE
      PSTRM2=0.
      PPOWM2=0.
      PPOWL2=0.
      P480B=0.
      PSMOT2=0.
      IF(TPH3.GE.6.) GO TO 1040
      PSTRM1=3606.
      PPOWM1=7.79
      PPOWL1=65.12
      P480A=94.5
      PSMOT1=8.
      GO TO 1050
 1040 CONTINUE
      PSTRM1=3844.
```

```
      PPOWM1=7.90
      PPOWL1=66.60
      P480A=109.5
      PSMOT1=8.
 1050 CONTINUE
      IF(NOSILO.GE.3) GO TO 1060
      PSTRM6=0.
      PPOWM6=0.
      PPOWL6=0.
      P480F=0.
      PSMOT6=0.
      GO TO 1070
 1060 CONTINUE
      PSTRM6=710.
      PPOWM6=1.84
      PPOWL6=16.28
      P480F=6.
      PSMOT6=2.
 1070 CONTINUE
      IF(NOPREP.GE.3) GO TO 1080
      PSTRM3=(NOPREP-1)*355.
      PPOWM3=(NOPREP-1)*0.920
      PPOWL3=(NOPREP-1)*8.14
      P480C=(NOPREP-1)*1.
      PSMOT3=(NOPREP-1)*1.
      GO TO 1090
 1080 CONTINUE
      PSTRM3=710.
      PPOWM3=1.84
      PPOWL3=16.28
      P480C=6.
      PSMOT3=2.
 1090 CONTINUE
      PSTRM4=(NOPREP+3)*355.+(NOPREP+1)*370.
      PPOWM4=(NOPREP+3)*.92+(NOPREP+1)*.96
      PPOWL4=((2*NOPREP)+4)*8.14
      P480D=(NOPREP+2)*3.5 +(1+NOPREP)*7.5
      PSMOT4=((2*NOPREP)+4)
      EHP(1)=2.
      EHP(2)=HPSLAK
      NOPMOT(1)=NOPREP
      NSPMOT(1)=NSPREP
 1100 CONTINUE
      PSTRM5=0.0
      PPOWM5=0.0
      PPOWL5=0.0
      P480E=0.0
      PSMOT5=0.0
      PSTRM7=0.0
      PPOWL7=0.0
      PPOWM7=0.0
      P480G =0.0
```

```
      PSMOT7=0.0
      IF(IADD.EQ.0) GO TO 1110
      PSTRM5=879.
      PPOWM5=1.84
      PPOWL5=16.28
      P480E=11.5
      PSMOT5=2
      PSTRM7=355.
      PPOWM7=0.92
      PPOWL7=8.14
      P480G =1.0
      PSMOT7=1
1110  CONTINUE
      EHP(3)=PMPHP(1)
      EHP(4)=FANHP
      EHP(5)=PMPHP(2)
      EHP(6)=PMPHP(3)
      EHP(7)=PMPHP(4)
      EHP(8)=PMPHP(5)
      EHP(9)=PMPHP(6)
      EHP(10)=PMPHP(7)
      EHP(11)=AGITHP(1)
      EHP(12)=AGITHP(9)
      EHP(13)=AGITHP(2)
      EHP(14)=AGITHP(8)
      EHP(15)=AGITHP(10)
      EHP(16)=COMPHP
      EHP(17)=AGITHP(3)
      EHP(18)=VERTHP(1)
      EHP(19)=VERTHP(2)
      EHP(20)=VERTHP(3)
      EHP(21)=VERTHP(4)
      EHP(22)=VERTHP(5)
      EHP(23)=PMPHP(8)
      EHP(24)=PMPHP(9)
      EHP(25)=PMPHP(10)
      EHP(26)=PMPHP(11)
      EHP(27)=AGITHP(5)
      EHP(28)=AGITHP(6)
      EHP(29)=RAKEHP
      EHP(30)=HPVAC
      EHP(31)=HPFDRV
      EHP(32)=HPFPMP
      EHP(33)=PCOVHP
      EHP(34)=HPFCCM
      EHP(35)=HPFIX(1)
      EHP(36)=HPFIX(2)
      EHP(37)=HPFIX(3)
      EHP(38)=HPFIX(4)
      EHP(39)=HPFIX(5)
      EHP(40)=HPFIX(6)
      EHP(41)=HPFIX(7)
```

```
      EHP(42)=HPFIX(8)
C     UNDEFINED ELECTRICAL VARIABLES
      NOPMOT(2)=NOPREP
      NSPMOT(2)=NSPREP
      NOPMOT(4)=NOTRAN
      NSPMOT(4)=NOREDN
      NOPMOT(11)=NOTNK(1)
      NSPMOT(11)=0
      NOPMOT(12)=NOTNK(9)
      NSPMOT(12)=0
      NOPMOT(13)=NOTRAN
      NSPMOT(13)=NOREDN
      IF (ISCRUB.EQ.4.OR.ISCRUB.EQ.6) NOPMOT(13)=0
      IF (ISCRUB.EQ.4.OR.ISCRUB.EQ.6) NSPMOT(13)=0
      NOPMOT(14)=NOTRAN
      NSPMOT(14)=NOREDN
      IF(ISCRUB.NE.3.OR.ISCRUB.NE.5) NOPMOT(14)=0
      IF(ISCRUB.NE.3.OR.ISCRUB.NE.5) NSPMOT(14)=0
      NOPMOT(15)=NOTNK(10)
      NSPMOT(15)=0
      NOPMOT(16)=NOTRAN
      IF(IFOX.LT.1) NOPMOT(16)=0
      NSPMOT(16)=NSCOMP
      NOPMOT(17)=NOTNK(3)
      NSPMOT(17)=0
      NOPMOT(27)=NOTNK(5)
      NSPMOT(27)=0
      NOPMOT(28)=NOTNK(6)
      NSPMOT(28)=0
      NOPMOT(29)=NOTHIK
      NSPMOT(29)=0
      NOPMOT(30)=NOFILT
      NSPMOT(30)=NSFILT
      NOPMOT(31)=NOFILT
      NSPMOT(31)=NSFILT
      NOPMOT(32)=NOFILT
      NSPMOT(32)=NSFILT
      IF(IADD.LT.1) NOPMOT(33)=0
      NSPMOT(33)=0
      IF(ISLUDG.LT.3) GO TO 1111
      NOPMOT(34)=1
      NSPMOT(34)=0
1111  IF(IFIXS.LT.1) GO TO 1112
      NOPMOT(35)=1
      NOPMOT(36)=1
      NSPMOT(36)=NSILFX-1
      NOPMOT(37)=1
      NSPMOT(37)=NASHS-1
      NOPMOT(38)=1
      NOPMOT(39)=1
      NSPMOT(39)=1
      NOPMOT(40)=1
```

```
      NSPMOT(40)=1
      NOPMOT(41)=1
      NOPMOT(42)=1
 1112 CONTINUE
      DO 1120 MA=1,8
      STRM(MA)=0.
      POWM(MA)=0.
      POWL(MA)=0.
      HP480(MA)=0.
      NSMOT(MA)=0
      NMMOT(MA)=0
      NLMOT(MA)=0
      ARHP(MA)=0.
      OPHP(MA)=0.
      SPHP(MA)=0.
      CTRMC(MA)=0.
      CTRLC(MA)=0.
      POWMMC(MA)=0.
      POWMLC(MA)=0.
      POWSMC(MA)=0.
      POWSLC(MA)=0.
      TRFRMC(MA)=0.
      TRFRLC(MA)=0.
      PSARHP(MA)=0.
 1120 CONTINUE
      MA = 1
      DO 1210 ME=1,42
      IF(ISLUDG.EQ.1.AND.ME.GT.19) GO TO 1210
      IF(ME.GT.1) MA = 2
      IF(ME.EQ.4) MA = 3
      IF(ME.GT.6) MA = 4
      IF(ME.EQ.9.OR.ME.EQ.12.OR.ME.EQ.15.OR.ME.EQ.16) MA = 5
      IF(ME.EQ.19) MA = 7
      IF(ME.GT.19) MA = 8
      IF(ME.EQ.33) MA = 2
      INOMOT(ME)=NOPMOT(ME)+NSPMOT(ME)
      DO 1130 IN=1,10
      IF(EHP(ME).LE.RHP(IN))GO TO 1140
 1130 CONTINUE
      IN=10
 1140 II=IN
      DO 1160 JJ=1,5
      IF(EHP(ME).EQ.0.0) GO TO 1150
      COST(JJ)=ELECT(II,JJ,1)*EHP(ME)**ELECT(II,JJ,2)
      GO TO 1160
 1150 COST(JJ)=0.0
 1160 CONTINUE
      STRM(MA)=COST(1)*INOMOT(ME)+STRM(MA)
      POWM(MA)=(COST(2)+COST(4))*INOMOT(ME)+POWM(MA)
      POWL(MA)=(COST(3)+10.*COST(4))*INOMOT(ME)+POWL(MA)
      IF(COST(5)-2.)1170,1180,1190
 1170 HP480(MA)=EHP(ME)*INOMOT(ME)+HP480(MA)
```

```
      NSMOT(MA)=INOMOT(ME)+NSMOT(MA)
      GO TO 1200
 1180 NMMOT(MA)=INOMOT(ME)+NMMOT(MA)
      GO TO 1200
 1190 NLMOT(MA)=INOMOT(ME)+NLMOT(MA)
 1200 ARHP(MA)=EHP(ME)*INOMOT(ME)+ARHP(MA)
      OPHP(MA)=EHP(ME)*NOPMOT(ME)+OPHP(MA)
      SPHP(MA)=EHP(ME)*NSPMOT(ME)+SPHP(MA)
 1210 CONTINUE
 1220 CONTINUE
      STRM(1)=STRM(1)+PSTRM1+PSTRM2+PSTRM7
      STRM(2)=STRM(2)+PSTRM3+PSTRM4+PSTRM5+PSTRM6
      POWM(1)=POWM(1)+PPOWM1+PPOWM2+PPOWM7
      POWM(2)=POWM(2)+PPOWM3+PPOWM4+PPOWM5+PPOWM6
      POWL(1)=POWL(1)+PPOWL1+PPOWL2+PPOWL7
      POWL(2)=POWL(2)+PPOWL3+PPOWL4+PPOWL5+PPOWL6
      PHPA=P480A+P480B+P480G
      PHPB=P480C+P480D+P480E+P480F
      HP480(1)=HP480(1)+PHPA
      HP480(2)=HP480(2)+PHPB
      NSMOT(1)=NSMOT(1)+PSMOT1+PSMOT2+PSMOT7
      NSMOT(2)=NSMOT(2)+PSMOT3+PSMOT4+PSMOT5+PSMOT6
      ARHP(1)=ARHP(1)+PHPA
      ARHP(2)=ARHP(2)+PHPB
      OPHP(1)=OPHP(1)+PHPA
      OPHP(2)=OPHP(2)+PHPB
      SBHP = 1.8 * 0.25 * (NSBF+NSBR)
C     ELECTRICAL POWER AND CONTROL DISTANCES
      PSARHP(1)=ARHP(1)+ARHP(2)
      PSARHP(3)=ARHP(1)+ARHP(2)+ARHP(3)+ARHP(4)+ARHP(5)+ARHP(6)+ARHP(7)
     1+ARHP(8)+SBHP
      PSARHP(7)=ARHP(7)
      PSARHP(8)=ARHP(7)+ARHP(8)
      PSDIST(1)=1500
      PSDIST(8) = 350.
      REMCTR(3)=250.
      REMCTR(1)=PSDIST(1) + REMCTR(3)
      REMCTR(2)=PSDIST(1) + REMCTR(3)
      REMCTR(4)=250.
      REMCTR(5)=250.
      REMCTR(6)=250.
      REMCTR(7)=DISTPD + REMCTR(3) + PSDIST(8)
      REMCTR(8)=PSDIST(8) + REMCTR(3)
      PSDIST(2)=0.0
      PSDIST(3)=500.
      PSDIST(4)=0.0
      PSDIST(5)=0.0
      PSDIST(6)=0.0
      PSDIST(7)=DISTPD
      AMOTDT(1)=350.
      AMOTDT(2)=200.
      AMOTDT(3)=250.
```

```
      AMOTDT(4)=250.
      AMOTDT(5)=250.
      AMOTDT(6)=250.
      AMOTDT(7)=100.
      AMOTDT(8)=350.
      NHEAT = 4 * NTOTAL
      NSUB = 8
      IF(ISCRUB.GT.2) NSUB = NSUB + 3
      IF(IRH.LT.1) NHEAT = 0
      NSUB = NSUB * NTOTAL
C ELECTRICAL MOTOR CONTROL COSTS
      DO1330 MA = 1,8
      NOTMOT(MA)=NSMOT(MA)+NMMOT(MA)+NLMOT(MA)
      NCARD = (NOTMOT(MA)-1)/20+1
      IF(MA.EQ.4) NCARD = (NOTMOT(4) + NSUB-1)/20+1
      IF(MA.EQ.6) NCARD = (NHEAT-1)/20+1
      CTRMC(MA)=(NSMOT(MA)+NMMOT(MA))*675.+NCARD*2.97*
     &REMCTR(MA)+NLMOT(MA)*(3340.+2.11*REMCTR(MA)+2.23*AMOTDT(MA))
      CTRLC(MA)=(NSMOT(MA)+NMMOT(MA))*81.+NCARD*21.35*
     &REMCTR(MA)+NLMOT(MA)*(147.+21.27*REMCTR(MA)+10.64*AMOTDT(MA))
C     ELECTRICAL MOTOR POWER COSTS
      POWMMC(MA)=STRM(MA)+NOTMOT(MA)*(480.+.80*AMOTDT(MA))+
     &POWM(MA)*AMOTDT(MA)
      POWMLC(MA)=.25*STRM(MA)+NOTMOT(MA)*(80.+6.83*AMOTDT(MA))+
     &POWL(MA)*AMOTDT(MA)
C     ELECTRICAL SUPPLY TO AREAS
      NCIRCB=2*IFIX((PSARHP(MA)-1.)/(2714.*5.)+1.)
      IF(NCIRCB.LT.1) NCIRCB=1
      NCABLE=(PSARHP(MA)-1)/2714+1
      IF(NCABLE.LT.1) NCABLE=1
      HPCABL=PSARHP(MA)/NCABLE
      IF(NCABLE.GT.8)GO TO 1280
      IF(PSARHP(MA).GT.470.)GO TO 1230
      POWSMC(MA)=1.10*PSDIST(MA)+1.17*PSDIST(MA)+48600.
      POWSLC(MA)=2.75*PSDIST(MA)+11.70*PSDIST(MA)+3400.
      GO TO 1290
1230  WIREMC(MA)=NCABLE*2.05060E-3*HPCABL**1.00048*PSDIST(MA)
      WIRELC(MA)=NCABLE*5.14005E-3*HPCABL**1.00048*PSDIST(MA)
      IF(PSARHP(MA).GT.300.)GO TO 1240
      CONDMC(MA)=1.17*PSDIST(MA)
      CONDLC(MA)=11.70*PSDIST(MA)
      GO TO 1250
1240  CONDMC(MA)=NCABLE*6.31298E-4*HPCABL**1.29600*PSDIST(MA)
      CONDLC(MA)=10.*CONDMC(MA)
1250  IF(PSARHP(MA).GT.8150.)GO TO 1260
      CIRBMC(MA)=48600.
      CIRBLC(MA)=3400.
      GO TO 1270
1260  CIRBMC(MA)=NCIRCB*29700.
      CIRBLC(MA)=NCIRCB*1700.
1270  POWSMC(MA)=WIREMC(MA)+CONDMC(MA)+CIRBMC(MA)
      POWSLC(MA)=WIRELC(MA)+CONDLC(MA)+CIRBLC(MA)
```

```
      GO TO 1290
C     WIRE AND CONDUIT AT 2714 KVA
 1280 POWSMC(MA)=NCABLE*21.66*PSDIST(MA)+NCIRCB*29700.
      POWSLC(MA)=NCABLE*170.52*PSDIST(MA)+NCIRCB*1700.
C     AREA TRANSFORMER COSTS
 1290 IF(HP480(MA).GT.10.)GO TO 1300
      TRFRMC(MA)=360.
      GO TO 1320
 1300 IF(HP480(MA).GT.75.)GO TO 1310
      TRFRMC(MA)= 750.
      GO TO 1320
 1310 TRFRMC(MA)=111.977*HP480(MA)**.75169
 1320 TRFRLC(MA)= 125.  + 0.10*TRFRMC(MA)
C     TOTAL ELECTRICAL AREA COSTS
      ELEMAT(MA)=(CTRMC(MA)+POWMMC(MA)+POWSMC(MA)+TRFRMC(MA))*RATMAT*1.1
      ELELAB(MA)=(CTRLC(MA)+POWMLC(MA)+POWSLC(MA)+TRFRLC(MA))*RATLAB*1.1
 1330 CONTINUE
      NSR = 0
      NSF =(NSBF/25) + 1
      IF(IRH.NE.0)NSR =(NSBR/25) + 1
      ESB4M =(15.5148*NSF+( NSBF * 0.20))*AMOTDT(4)* RATMAT* 1.15
      ESB6M =(15.5148*NSR+(NSBR*0.20))*AMOTDT(6)*RATMAT* 1.15
      ESB4L =(155.148*NSF+NSBF*0.92)*AMOTDT(4)*RATLAB* 1.15
      ESB6L = (155.148*NSR+NSBR*0.92)*AMOTDT(6)*RATLAB* 1.15
      ELEMAT(4) =  ELEMAT(4) + ESB4M
      ELEMAT(6) =  ELEMAT(6) + ESB6M
      ELELAB(4) =  ELELAB(4) + ESB4L
      ELELAB(6) =  ELELAB(6) + ESB6L
      IF(IFOX.GT.0) GO TO 1335
      ELEMAT(5) = 0.0
      ELELAB(5) = 0.0
 1335 CONTINUE
      IF(IRH.GT.0) GO TO 1340
      ELEMAT(6) = 0.0
      ELELAB(6) = 0.0
 1340 CONTINUE
      ELEMAT(7)=ELEMAT(7)+ELEMAT(8)
      ELELAB(7)=ELELAB(7)+ELELAB(8)
      ELEMAT(8)=0.0
      ELELAB(8)=0.0
      REVHP=OPHP(1)+OPHP(2)+OPHP(3)+OPHP(4)+OPHP(5)+OPHP(6)+OPHP(7)+
     1OPHP(8)+SBHP
      RETURN
      END
```

```
$DEBUG
$NOFLOATCALLS
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM TOTALS
C     REVISION DATE 84-03-13
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE TOTALS
     1       (SERV,REVHP,ARCHEC,ARLAND,CONEXP,
     2       CONTIN,CONTRA,DIRECT,DIREXC,DUCLAB,DUCMAT,
     3       ELELAB,ELEMAT,ENGDES,ENSLAB,ENSMAT,
     4       ENTCON,PIPLAB,PIPMAT,PONDIR,PONIND,ROYALS,SALTAX,
     6       STRTUP,SUBCAP,SUBFIX,TAXEXC,TAXPND,
     7       TOTINV,TOTPCP,TOTPRC,WORCAP)
      COMMON/DUM/VERGPM(5),VERHD(5),VTOTLC(5),VTOTMC(5),VERTHP(8),
     &    AGITHP(10),AGITLC(10),AGITMC(10),DIATNK(10),HTTNK(10),
     &    TNKLC(10),TNKMC(10),VOLTNK(10),GPMPMP(11),PMPHP(11),
     &    TPMPLC(11),TPMPMC(11),TPH3,HPVAC,THKHT,CMECLC,CMECMC,CON3LG,
     &    CON8LG,DIASIL,DISPHD,FILTAR,HPMILL,HPSLAK,HTELE2,HTSILS,
     &    RAKEHP,REHTLC,REHTMC,SOBLLC,SOBLMC,SOBM,TFILTL,TFILTM,THIKAR,
     &    THKDIA,TPHSLK,TOTTKL,TOTTKM,VOLSIL,SOBL,NOTNK(10),NOMOT(45),
     &    NOPMOT(45),NSPMOT(45),NGR18,NOFILT,NOPREP,NOSILO,NOTHIK,
     &    NSFILT
      COMMON/DUMM/PMPFDN(11),TNKFDN(10),SPGR(10),BLDLAB(9),BLDMAT(9),
     1 EQULAB(9),EQUMAT(9),FDNLAB(9),FDNMAT(9),STRLAB(9),STRMAT(9),
     2 FBLDSF,FILFDL,FILFDM,HPFDRV,HPFPMP,VSCTNK,VERFDN,ELE2L,ELE2M,
     3 CSLAKL,CSLAKM,GPMILL,PCONL1,PCONL2,PCONL4,PCONM1,PCONM2,PCONM4,
     4 PEQLC1,PEQLC2,PEQMC1,PEQMC2,PSTLL1,PSTLL2,PSTLL3,PSTLL4,PSTLM1,
     5 PSTLM2,PSTLM3,PSTLM4,TPSLAK,TSLKHP,CMILLL,CMILLM,GPMTRN,NSUMPR
C
      COMMON/BOILER/MW,BHR,EXSAIR,RMW,PREC,SPASH,WPRITE,KCLEAN,SMRW,SMCL
      COMMON/FLOWS/GPM,MFLOW,WFLOW,TOTFLO
      COMMON/IND/ENGIN,ARCTEC,FLDEXP,FEES,CONT,START,CONINT,FILMAT(9)
      COMMON/LACOS3/WTSM,WTSL,WTSL1,TLAB,CRATE,ZRATE,EQUIPC,NDOZER,NCOMP
      COMMON/PCOST/POND(9),PDACRE,TOPHRS,PNDCAP,PNDLAB(9),PNDMAT(9),ELAN
      COMMON/PNDIND/INDPND,PENGIN,PARCH,PFLDEX,PFEES,PCONT,PSTART,PCONIN
      COMMON/RRIND/XINT,PCTMNT,PDMNTP,XINFLA,PCTOVR,PCTADM,UNDCAP,
     1             PCTINS,UC,IECON
      COMMON/RTIME/TR,ALCONT(9)
      COMMON/TAXES/ITAXFR,TXFRRT,TXRAT,FRRAT,SERVRT,ROYALT,IOTIME,OTRATE
      REAL MW
      REAL MFLOW(20,12),WFLOW(20,12),TOTFLO(20),GPM(20)
      REAL*8 DSBCAP,DALAND,DWRCAP,DTOFLO,MGWATT,KBTU,DLBH3,
     &DWLBH3,DWPS,AQ(18),DDIREC,DPOND
      REAL *8 XINT,PCTMNT,PDMNTP,XINFLA,PCTOVR,PCTADM,UNDCAP,PCTINS,UC(1
     1        8)
C
      DIMENSION PIPMAT(9),PIPLAB(9),DUCMAT(9),
     &DUCLAB(9),ELEMAT(9),ELELAB(9),ENSMAT(9),ENSLAB(9),
```

```
     &SERV(9),DIRECT(9),DIRIND(9),ENGDES(9),CONEXP(9),CONTRA(9),
     &CONTIN(9),SUBFIX(9),STRTUP(9),ENTCON(9),SUBCAP(9),ARLAND(9),
     &WORCAP(9),TOTINV(9),DIREXC(9),TAXLAN(9)
      DIMENSION TOTMAT(9),TOTLAB(9),SALTAX(9),TOTPRC(9),TOTPCP(9),
     1          PENGDE(9),PCONEX(9),PCONTR(9),PIRIND(9),PONTIN(9),
     2          PNDFIX(9),PSTRTU(9),PENTCO(9),PONCAP(9),PONIND(9),
     3          ARCHEC(9),PARCHE(9),ROYALS(9)
      DIMENSION PONDIR(9),TAXPND(9),TAXEXC(9),FILMA2(9)
C
C     PROCEDURE FOR DISPLAYING INVESTMENT
      DO 999 N = 1,9
      PONIND(N) = 0.0
      TOTINV(N) = 0.0
      FILMA2(N) = 0.0
      ALCONT(N) = 0.0
      SALTAX(N) = 0.0
      TAXPND(N) = 0.0
      TAXLAN(N) = 0.0
      TAXEXC(N) = 0.0
      TOTPRC(N) = 0.0
      SERV(N) = 0.0
      ROYALS(N) = 0.0
      TOTPCP(N) = 0.0
      DIREXC(N) = 0.0
      DIRECT(N) = 0.0
      PONDIR(N) = 0.0
      DIRIND(N) = 0.0
      PENGDE(N) = 0.0
      PARCHE(N) = 0.0
      PCONEX(N) = 0.0
      PCONTR(N) = 0.0
      PIRIND(N) = 0.0
      PONDIR(N) = 0.0
      CONTIN(N) = 0.0
      PONTIN(N) = 0.0
      SUBFIX(N) = 0.0
      PNDFIX(N) = 0.0
      STRTUP(N) = 0.0
      ENTCON(N) = 0.0
      SUBCAP(N) = 0.0
      PSTRTU(N) = 0.0
      PENTCO(N) = 0.0
      PONCAP(N) = 0.0
      ENGDES(N) = 0.0
      ARCHEC(N) = 0.0
      CONEXP(N) = 0.0
      CONTRA(N) = 0.0
999   CONTINUE
      FILMAT(7) = ELAN+WTSL
      FILMA2(7) = ELAN
      DO 1000 K=1,7
      EQUMAT(K)=EQUMAT(K)/1000.
```

```
      EQULAB(K)=EQULAB(K)/1000.
      PIPMAT(K)=PIPMAT(K)/1000.
      PIPLAB(K)=PIPLAB(K)/1000.
      DUCMAT(K)=DUCMAT(K)/1000.
      DUCLAB(K)=DUCLAB(K)/1000.
      FDNMAT(K)=FDNMAT(K)/1000.
      FDNLAB(K)=FDNLAB(K)/1000.
      PNDLAB(K)=PNDLAB(K)/1000.
      PNDMAT(K)=PNDMAT(K)/1000.
      ALCONT(K)=PNDMAT(K)+PNDLAB(K)
      POND(K)=POND(K)/1000.
      STRMAT(K)=STRMAT(K)/1000.
      STRLAB(K)=STRLAB(K)/1000.
      ELEMAT(K)=ELEMAT(K)/1000.
      ELELAB(K)=ELELAB(K)/1000.
      ENSMAT(K)=ENSMAT(K)/1000.
      ENSLAB(K)=ENSLAB(K)/1000.
      BLDMAT(K)=BLDMAT(K)/1000.
      BLDLAB(K)=BLDLAB(K)/1000.
      ARLAND(K)=ARLAND(K)/1000.
      FILMA2(K)=FILMA2(K)/1000.
      FILMAT(K)=FILMAT(K)/1000.
 1000 CONTINUE
C     PROCEDURE FOR SUMMING AREA INVESTMENT COSTS
C
      DO 1060 I=1,7
      TOTMAT(I)=EQUMAT(I)+PIPMAT(I)+DUCMAT(I)+FDNMAT(I)+
     1          STRMAT(I)+ELEMAT(I)+ENSMAT(I)+BLDMAT(I)
      TOTLAB(I)=EQULAB(I)+PIPLAB(I)+DUCLAB(I)+FDNLAB(I)+
     1          STRLAB(I)+ELELAB(I)+ENSLAB(I)+BLDLAB(I)
      IF(ITAXFR.NE.1)  GO TO 1010
      TAXPND(I)=(PNDMAT(I)+FILMA2(I))*TXFRRT/100.0
      TAXLAN(I)=FILMA2(I)*TXFRRT/100.0
      TAXEXC(I)=TOTMAT(I)*TXFRRT/100.0
      SALTAX(I)=TAXPND(I)+TAXEXC(I)
 1010 CONTINUE
      TOTPRC(I)=TOTMAT(I)+TOTLAB(I)+TAXEXC(I)
      SERV(I)=TOTPRC(I)*SERVRT/100.0
      ROYALS(I)=TOTPRC(I)*ROYALT/100.0
      TOTPCP(I)=TOTPRC(I)+POND(I)+TAXPND(I)+FILMAT(I)
      DIREXC(I)=TOTPRC(I)+SERV(I)
      WRITE(8,*)  ' I,TOTMAT,TOTLAB,TAXEXC,SERV',
     &            I,TOTMAT(I),TOTLAB(I),TAXEXC(I),SERV(I)
      DIRECT(I)=TOTPCP(I)+SERV(I)
      PONDIR(I)=POND(I)+TAXPND(I)-TAXLAN(I)
      ENGDES(I)=DIREXC(I)*ENGIN/100.0
      ARCHEC(I)=DIREXC(I)*ARCTEC/100.0
      CONEXP(I)=DIREXC(I)*FLDEXP/100.0
      CONTRA(I)=DIREXC(I)*FEES/100.0
      DIRIND(I)=DIREXC(I)+ENGDES(I)+ARCHEC(I)+CONEXP(I)+CONTRA(I)
      PENGDE(I)=PONDIR(I)*PENGIN/100.0
      PARCHE(I)=PONDIR(I)*PARCH/100.0
```

```
      PCONEX(I)=PONDIR(I)*PFLDEX/100.0
      PCONTR(I)=PONDIR(I)*PFEES/100.0
      PIRIND(I)=PONDIR(I)+PENGDE(I)+PARCHE(I)+PCONEX(I)+PCONTR(I)
      PONDIR(I)=PONDIR(I)+TAXLAN(I)+FILMAT(I)
      IF(IECON.EQ.1) GO TO 1020
      CONTIN(I)=DIREXC(I)*CONT/100.0
      PONTIN(I)=PONDIR(I)*PCONT/100.0
      GO TO 1030
 1020 CONTINUE
      WRITE (8,*) ' DIRIND,CONT,I=',DIRIND(I),CONT,I
      CONTIN(I)=DIRIND(I)*CONT/100.0
      PONTIN(I)=(PIRIND(I)+TAXLAN(I)+FILMAT(I))*PCONT/100.0
 1030 CONTINUE
      SUBFIX(I)=DIRIND(I)+CONTIN(I)
      PNDFIX(I)=PIRIND(I)+PONTIN(I)
      STRTUP(I)=SUBFIX(I)*START/100.0
      ENTCON(I)=SUBFIX(I)*CONINT/100.0
      SUBCAP(I)=SUBFIX(I)+STRTUP(I)+ENTCON(I)
      PSTRTU(I)=PNDFIX(I)*PSTART/100.0
      PENTCO(I) =(PNDFIX(I)+TAXLAN(I)+FILMAT(I)) * PCONIN/100.0
      PONCAP(I)=PNDFIX(I)+PSTRTU(I)+PENTCO(I)+TAXLAN(I)+FILMAT(I)
      IF(INDPND.EQ.1) GO TO 1040
      ENGDES(I)=ENGDES(I)+PENGDE(I)
      ARCHEC(I)=ARCHEC(I)+PARCHE(I)
      CONEXP(I)=CONEXP(I)+PCONEX(I)
      CONTRA(I)=CONTRA(I)+PCONTR(I)
      CONTIN(I)=CONTIN(I)+PONTIN(I)
      GO TO 1050
 1040 CONTINUE
      PONIND(I)=PENGDE(I)+PARCHE(I)+PCONEX(I)+PCONTR(I)+PONTIN(I)
 1050 CONTINUE
      SUBFIX(I)=SUBFIX(I)+PNDFIX(I)+FILMAT(I)+TAXLAN(I)
      STRTUP(I)=STRTUP(I)+PSTRTU(I)
      ENTCON(I)=ENTCON(I)+PENTCO(I)
      SUBCAP(I)=SUBCAP(I)+PONCAP(I)
 1060 CONTINUE
C
      DO 1065 I2 = 1,7
      EQUMAT(8)=EQUMAT(I2)+EQUMAT(8)
      EQULAB(8)=EQULAB(I2)+EQULAB(8)
      PIPMAT(8)=PIPMAT(I2)+PIPMAT(8)
      PIPLAB(8)=PIPLAB(I2)+PIPLAB(8)
      DUCMAT(8)=DUCMAT(I2)+DUCMAT(8)
      DUCLAB(8)=DUCLAB(I2)+DUCLAB(8)
      FDNMAT(8)=FDNMAT(I2)+FDNMAT(8)
      FDNLAB(8)=FDNLAB(I2)+FDNLAB(8)
      PNDLAB(8)=PNDLAB(I2)+PNDLAB(8)
      PNDMAT(8)=PNDMAT(I2)+PNDMAT(8)
      ALCONT(8)=ALCONT(I2)+ALCONT(8)
      POND(8)=POND(I2)+POND(8)
      STRMAT(8)=STRMAT(I2)+STRMAT(8)
      STRLAB(8)=STRLAB(I2)+STRLAB(8)
```

```
      ELEMAT(8)=ELEMAT(I2)+ELEMAT(8)
      ELELAB(8)=ELELAB(I2)+ELELAB(8)
      ENSMAT(8)=ENSMAT(I2)+ENSMAT(8)
      ENSLAB(8)=ENSLAB(I2)+ENSLAB(8)
      BLDMAT(8)=BLDMAT(I2)+BLDMAT(8)
      BLDLAB(8)=BLDLAB(I2)+BLDLAB(8)
      TAXEXC(8)=TAXEXC(I2)+TAXEXC(8)
      TAXPND(8)=TAXPND(I2)+TAXPND(8)
      TAXLAN(8)=TAXLAN(I2)+TAXLAN(8)
      SALTAX(8)=SALTAX(I2)+SALTAX(8)
      TOTPCP(8)=TOTPCP(I2)+TOTPCP(8)
      TOTPRC(8)=TOTPRC(I2)+TOTPRC(8)
      SERV(8)=SERV(I2)+SERV(8)
      DIREXC(8)=DIREXC(I2)+DIREXC(8)
      PONDIR(8)=PONDIR(I2)+PONDIR(8)
      FILMAT(8)=FILMAT(I2)+FILMAT(8)
      DIRECT(8)=DIRECT(I2)+DIRECT(8)
      ENGDES(8)=ENGDES(I2)+ENGDES(8)
      ARCHEC(8)=ARCHEC(I2)+ARCHEC(8)
      CONEXP(8)=CONEXP(I2)+CONEXP(8)
      CONTRA(8)=CONTRA(I2)+CONTRA(8)
      CONTIN(8)=CONTIN(I2)+CONTIN(8)
      PONIND(8)=PONIND(I2)+PONIND(8)
      SUBFIX(8)=SUBFIX(I2)+SUBFIX(8)
      STRTUP(8)=STRTUP(I2)+STRTUP(8)
      ENTCON(8)=ENTCON(I2)+ENTCON(8)
      SUBCAP(8)=SUBCAP(I2)+SUBCAP(8)
      ROYALS(8)=ROYALS(I2)+ROYALS(8)
      ARLAND(8)=ARLAND(I2)+ARLAND(8)
      DPOND = (PONDIR(8)) * 1000.
 1065 CONTINUE
      CALL WORKCP(REVHP,DIRECT,WORCAP)
      DO 1070 I=1,7
      TOTINV(I)=SUBCAP(I)+ROYALS(I)+ARLAND(I)+WORCAP(I)
 1070 CONTINUE
      DO 1066 I3 = 1,7
      TOTINV(8)=TOTINV(I3)+TOTINV(8)
 1066 CONTINUE
C     XPCTDR=100./DIRECT(8)
      XPCTDR=1.0/RMW
      EQUMAT(9)=EQUMAT(8)*XPCTDR
      EQULAB(9)=EQULAB(8)*XPCTDR
      PIPMAT(9)=PIPMAT(8)*XPCTDR
      PIPLAB(9)=PIPLAB(8)*XPCTDR
      DUCMAT(9)=DUCMAT(8)*XPCTDR
      DUCLAB(9)=DUCLAB(8)*XPCTDR
      FDNMAT(9)=FDNMAT(8)*XPCTDR
      FDNLAB(9)=FDNLAB(8)*XPCTDR
      PNDLAB(9)=PNDLAB(8)*XPCTDR
      PNDMAT(9)=PNDMAT(8)*XPCTDR
      ALCONT(9)=ALCONT(8)*XPCTDR
      POND(9)=POND(8)*XPCTDR
```

```
      STRMAT(9)=STRMAT(8)*XPCTDR
      STRLAB(9)=STRLAB(8)*XPCTDR
      ELEMAT(9)=ELEMAT(8)*XPCTDR
      ELELAB(9)=ELELAB(8)*XPCTDR
      ENSMAT(9)=ENSMAT(8)*XPCTDR
      ENSLAB(9)=ENSLAB(8)*XPCTDR
      BLDMAT(9)=BLDMAT(8)*XPCTDR
      BLDLAB(9)=BLDLAB(8)*XPCTDR
      TAXEXC(9)=TAXEXC(8)*XPCTDR
      TAXPND(9)=TAXPND(8)*XPCTDR
      SALTAX(9)=SALTAX(8)*XPCTDR
      TOTPCP(9)=TOTPCP(8)*XPCTDR
      TOTPRC(9)=TOTPRC(8)*XPCTDR
      SERV(9)=SERV(8)*XPCTDR
      DIREXC(9)=DIREXC(8)*XPCTDR
      FILMAT(9)=FILMAT(8)*XPCTDR
      PONDIR(9)=PONDIR(8)*XPCTDR
      DIRECT(9)=DIRECT(8)*XPCTDR
      ENGDES(9)=ENGDES(8)*XPCTDR
      ARCHEC(9)=ARCHEC(8)*XPCTDR
      CONEXP(9)=CONEXP(8)*XPCTDR
      CONTRA(9)=CONTRA(8)*XPCTDR
      CONTIN(9)=CONTIN(8)*XPCTDR
      PONIND(9)=PONIND(8)*XPCTDR
      SUBFIX(9)=SUBFIX(8)*XPCTDR
      STRTUP(9)=STRTUP(8)*XPCTDR
      ENTCON(9)=ENTCON(8)*XPCTDR
      SUBCAP(9)=SUBCAP(8)*XPCTDR
      ROYALS(9)=ROYALS(8)*XPCTDR
      ARLAND(9)=ARLAND(8)*XPCTDR
      WORCAP(9)=WORCAP(8)*XPCTDR
      TOTINV(9)=TOTINV(8)*XPCTDR
C     PREMSIS LANDFILL MAINTENANCE DOES NOT INCLUDE EQUIPMENT, HENCE
C     TAX-FREIGHT IS REMOVED.  IF INCLUDED REPLACE TAXLAN-FILMAT
      DDIREC=DIRECT(8)*1000.
      DSBCAP=SUBCAP(8)*1000.
      DALAND=ARLAND(8) * 1000.
      DWRCAP=WORCAP(8) * 1000.
      RETURN
      END
```

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM TO ZERO VARIABLES
C     REVISION DATE 83-04-01
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE ZERO
C
      COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
     1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
     2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
     3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
     4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
     5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
     6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
      COMMON/ALKASH/ASHCAO, ASHMGO, SRCAO, SRMGO, SRTOT
      COMMON/BYPAS2/BYLBH(6), BYLBMH(6), BSCFM, BACFM, PASSBY, XPASSD, IBY
      COMMON/BOILER/MW, BHR, EXSAIR, RMW, PREC, SPASH, WPRITE, KCLEAN, SMRW, SMCL
      REAL MW
      COMMON/CID/CASEID(20)
      COMMON/COAL/HVC, WPC, WPH, WPO, WPN, WPSUL, WPCL, WPASH, WPH2O, SULO, ASHO
      COMMON/COMPLE/ICOMPL, IRPT, IRUN, NEXT, IEQPR, IWTBAL, ITM, INPOPT, NOPART
      COMMON/CONS/KGTA, KGEHT, KPCO2, C1, C2, C3, C4, PC1, PC2, PC3, PC4, PC5
      REAL KGTA, KGEHT, KPCO2
      COMMON/EMIT/ISO2, XSO2, ISSO2, ELISO2, SO2REF, SO2LMB, SO2PPM
      COMMON/FILCON/FCCTLC, FCCTMC, HPFCCM, LNHFCC, LNIFCC
      COMMON/FIXCOS/FIXADD, DS, TFIXEQ, TFIXLB, HPFIX(8)
      COMMON/FIXFDN/ALCONY, FDNICS, PUGFDN, NSILFX, NASHS
      COMMON/FIXSLO/VOLIMS, DIALIS, HTSIDL, VOLAHS, DIAASH, HTSIDA, TPHMIX
      COMMON/FLOWS/GPM, MFLOW, WFLOW, TOTFLO
      REAL MFLOW(20,12), WFLOW(20,12), TOTFLO(20), GPM(20)
      COMMON/FLYASH/IASH, ASHUPS, ASHSCR, ASHPRU, ASHPRS, ASHLBU, ASHLBS
      COMMON/HGAS/ACFM, THG, YCO2, YHCL, YSO2, YO2, YN2, YH2O, ASH, WCL, WASH, IESP
      COMMON/HGAS2/MPCT, LBMH, LBH, SCFM, HHG, HWG, CFR, BCFR, RCFR
      REAL MPCT(6), LBMH(6), LBH(6)
      COMMON/H2OAMT/SEEPRT, EVAPRT, RAIN, SEEP, EVAP, EVAPC, RAFAL, RAFALC,
     1              WLSM, H2OHYD, TWA, TWR
      COMMON/IND/ENGIN, ARCTEC, FLDEXP, FEES, CONT, START, CONINT, FILMAT(9)
      COMMON/INPT/JINPUT, JBC, JALK, JSSVAR,  JSRHT
      COMMON/JP/JP1, JP2, JP3, JP4, JP5, JP6, JP7, JP8, JP9, JP10
      COMMON/LACOS3/WTSM, WTSL, WTSL1, TLAB, CRATE, ZRATE, NDOZER, NCOMP
      COMMON/LANDIS/FILAND, TWLF, DISFUL, ELECLF, WATERF, FILANL, OVLF
      REAL LGMAX
      COMMON/LIQ2/IW4, AD, ADDC, ADLOSS, AMGLOS
      COMMON/LIQDIS/H2O, CAPPM, MGPPM, SO3PPM, SO4PPM, CLPPM, TDS, PHPOND, DGPM
      REAL MGPPM
      COMMON/LS/IALK, ISR, SR, SRADD, WPMGO, WPI, WPM, LSADD, LIMEAD, MGOADD, WMG
      REAL LSADD, LIMEAD, MGOADD
      COMMON/OPPSCH/IOPSCH, IYROP, IA(50)
      COMMON/OUTPT/JOUTPT, JIT, JEQ1, JEQ2, JEQ3
      COMMON/OUTPT1/JHGAS, JWGAS, JRAIR, JRGAS, JSRHO, JSKGAS
```

```
      COMMON/PCOST/POND(9), PDACRE, TOPHRS, PNDCAP, PNDLAB(9), PNDMAT(9), ELAN
      COMMON/PND1/PDEPTH, PMXEXC, DISTPD, ILINER, CLAYIN, CLAYUC, HYPLMC,
     1            HYPLLC, PSAMAX, ACREC, APRAIN, SAPILE
      COMMON/PT/ICONV, PHSTAR, CBSTAR, CB, CBR, HCO3EQ, SO3EQ, HSO3EQ
      COMMON/REHEET/OIL, OILGPM, HDUTY, RHSCFM, RDUST, RWDUST,
     &              RMPCT(6), RHLBMH(6), RHLBH(6)
      COMMON/RHAIR/ASCFM, ALBMH(6), ALBH(6)
      COMMON/RRIND/XINT, PCTMNT, PDMNTP, XINFLA, PCTOVR, PCTADM, UNDCAP,
     1             PCTINS, UC, IECON
      REAL *8 XINT, PCTMNT, PDMNTP, XINFLA, PCTOVR, PCTADM, UNDCAP, PCTINS, UC(1
     1       8)
      COMMON/RTIME/TR, ALCONT(9)
      COMMON/SCRUB/ISCRUB, IFOX, IADD, VLG, VTR, IFAN
      COMMON/SCRUBC/CSCRUB(14), SCRUBL, SCRUBM
      COMMON/SCRUBR/LG, V, PPMOUT, SO2R, ELIMIT, PR, PD, WPS, OX, PH, NSTAGE,
     1              NGRID, HS, ACS, ACSTRN
      REAL LG
      COMMON/SKGAS/SSCFM, SACFM, TSK, SDUST, SWDUST, SMPCT, SLBMH, SLPH, TGAS
      REAL SMPCT(6), SLBMH(6), SLBH(6)
      COMMON/SOLDIS/CASO3, CASO4, CACO3, INSOLS, WDIS, PSD, DENS, DEND
      REAL INSOLS
      COMMON/SRHT/TSTEAM, HVS, VRH, WSTEAM, ARH, DPRH, URH, Q, NT, NB, PL, OD,
     1            ARINCL, ARCORT, NINCLT, NCORTT, NINCLB, NCORTB, TEMPS
      DIMENSION ARH(6), DPRH(6), URH(6), NT(6), NB(6), OD(6), ARINCL(6),
     1          ARCORT(6), NINCLT(6), NCORTT(6), NINCLB(6), NCORTB(6)
      COMMON/TRAN/NOTRAN, NOREDN, NTOTAL, PCNTRN, NORAW, NOUTIL, NOUNIT,
     1            NOPROD, NOSHIP, NSPREP, NSCOMP, NOCOMP, NOBYDU
      COMMON/WGAS/WACFM, WSCFM, TWG, ASHOUT, WASHOT, WMPCT, WLBMH, WLBH
      REAL WMPCT(6), WLBMH(6), WLBH(6)
      COMMON/WGCOMP/YWCO2, YWHCL, YWSO2, YWO2, YWN2, YWH2O, WVAPOR, NTRAIN
      REAL NTRAIN
      COMMON/WRIT9/DSBCAP, DALAND, DWRCAP, DTOFLO, MGWATT, KBTU, DLBH3,
     1             DWLBH3, DWPS, AQ, DDIREC, DPOND, IHVC
      REAL *8 DSBCAP, DALAND, DWRCAP, DTOFLO, MGWATT, KBTU, DLBH3,
     1        DWLBH3, DWPS, AQ(40), DDIREC, DPOND
C
      COMMON/PCOSTS/PCLEAR, PDEXEC, SODM, SODL, PERIML, PERIMM, ROADMM, ROADLL,
     1 RECLAM
      COMMON/PNDARW/AB, AID, AIE, AOD, AS, DL, P, VEC, VETT, WELLM, WELLL, WPSITE
      COMMON/PNDIND/INDPND, PENGIN, PARCH, PFLDEX, PFEES, PCONT, PSTART, PCONIN
C     FIXATION
      DSBCAP=0.0
      DALAND=0.0
      DWRCAP=0.0
      DDIREC=0.0
      DPOND=0.0
      SKLAND = 0.0
      DRAINM = 0.0
      DRAINL = 0.0
      CLAYSK = 0.0
      DS = 0.0
      ALCONY = 0.0
```

```
      FDNICS = 0.0
      PUGFDN = 0.0
      NSILFX = 0
      NASHS = 0
      VOLIMS = 0.0
      DIALIS = 0.0
      ENGIN= 0.0
      ARCTEC= 0.0
      FLDEXP= 0.0
      FEES= 0.0
      CONT= 0.0
      START= 0.0
      CONINT= 0.0
      DO 64 I=1,9
   64 FILMAT(I) = 0.0
      HTSIDL = 0.0
      VOLAHS = 0.0
      DIAASH = 0.0
      HTSIDA = 0.0
      TPHMIX = 0.0
      DO 1 I=1,40
      AQ(I)=0.0
    1 CONTINUE
      DO 111 I = 1,8
      HPFIX(I) = 0.0
  111 CONTINUE
C     LANDFILL
      TWLF = 0.0
      ITM = 0
      WTSL = 0.0
      FILAND = 0.0
      DISFUL = 0.0
      ELECLF = 0.0
      WATERF = 0.0
      FILANL = 0.0
      OVLF = 0.0
C     BYPASS
      NOBYDU = 0
      BSCFM = 0.0
      BACFM = 0.0
      PASSBY = 0.0
      XPASSD = 0.0
      IBY = 0
      DO 1000 I = 1,6
      BYLBH(I) = 0.0
      BYLBMH(I) = 0.0
 1000 CONTINUE
      APRAIN=0.0
      EVAP=0.0
      EVAPC=0.0
      FCCTLC=0.0
      FCCTMC=0.0
```

```
      FILTSA=0.0
      HPFCCM=0.0
      ICOMPL=0
      IFLOOD=0
      LNHFCC=0
      LNIFCC=0
      RAFAL=0.0
      RAFALC=0.0
      SEEP=0.0
      TOPHRS=0.0
      WSTEAM=0.0
      AMGLOS=0.0
      ADLOSS=0.0
C
C     DISPOSAL COSTS
C
      WELLM=5000.
      WELLL=5100.
      RECLAM=3100.
      PCLEAR=1950.
      PDEXEC=1.75
      SODM=0.30
      SODL=0.49
      PERIML=6.05
      PERIMM=7.00
      ROADMM=6.00
      ROADLL=8.50
      CLAYUC = 0.0
      CLAYIN = 0.0
      HYPLMC = 0.0
      HYPLLC = 0.0
C
      DO 1010 I=1,9
      PNDLAB(I)=0.0
      PNDMAT(I)=0.0
      POND(I)=0.0
      FILMAT(I)=0.0
 1010 CONTINUE
      DO 1020 I=1,6
      DPRH(I)=0.0
      MPCT(I)=0.0
 1020 CONTINUE
      DO 1030 I=1,14
      CSCRUB(I)=0.0
 1030 CONTINUE
      DO 1040 I=1,20
      GPM(I)=0.0
      TOTFLO(I)=0.0
 1040 CONTINUE
      XINPUT = 1
      XBC = 1
      XALK = 1
```

```
XSSV = 1
XSRHT = 1
OUTPUT = 1
XHGAS = 1
XWGAS = 1
XRAIR = 1
XRGAS = 1
XSRHO = 1
XSKGAS = 1
XSSO = 1
XDIS = 1
XSTR = 1
XGPM = 1
XIT  = 1
IRPT = 1
IEQPR = 1
IWTBAL = 1
NOPART = 1
KCLEAN = 0
PREC = 0
SPASH = 0
WPRITE = 0
SMRW = 0
SMCL = 0
ASHCLN = 0
HVCLN = 0
WPC = 0
WPH = 0
WPN = 0
SULO = 0
ASHO = 0
ASHSCR = 0
VO2 = 0
VN2 = 0
SCFM = 0
WASH = 0
XLG = 0
ASHCAO = 0
IECON = 0
XINDEX = 0
YINDEX = 0
INVYR = 0
IREVYR = 0
BAGDLP = 0
BAGRAT = 0
BGCOST = 0
BGLIFE = 0
EFFPS = 0
ESPDLP = 0
RESIST = 0
SCARAT = 0
ICEPYE = 0
```

```
PCTADM = 0
UNDCAP = 0
PCTINS = 0
RETURN
END
```

```
$NOTSTRICT
$DEBUG
$LARGE
      SUBROUTINE TVAFGD
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     DSN=INVEST.FORT
C     PROGRAM REVISION DATE 84-03-13
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     FT02F001  REVENUE DATA FILE
C     FT03F001  MESSAGE PRINT FILE
C     FT05F001  INPUT READ CARDS UNIT
C     FT06F001  PRINT UNIT FILE
C     FT08F001  EDIT DATA FILE
C     FT09F001  COAL CLEAN FILE
C     BELOW IS LISTED THE JCL NECESSARY TO COMPILE ON THE TVA SYSTEM
C     ALL LINES ARE INSERTED IN COLUMN TWO TO ALLOW 'C' IN COLUMN ONE
C     COMPILES ON THE TVA SYSTEM 36 - 40 SECONDS
C
C//TXFTG1CL JOB 123456,FSUDHOFF.CEBM.2814,MSGLEVEL=1,CLASS=H,
C//            NOTIFY=CLALQ01
C//*MAIN  ORG=RGROUP03
C//*
C//PROCLIB  DD  DSN=L.SG508009.PROCLIB,DISP=SHR
C//COMPRESS EXEC PGM=IEBCOPY,REGION=100K
C//SYSPRINT DD SYSOUT=A
C//INOUT5 DD DSN=CLALQ01.INVEST.LOAD,DISP=(OLD,PASS)
C//SYSUT3   DD UNIT=SYSDA,SPACE=(TRK,(50,10))
C//SYSUT4   DD UNIT=SYSDA,SPACE=(TRK,(50,10))
C//SYSIN    DD *
C     COPY OUTDD=INOUT5,INDD=INOUT5
C//*
C//GFORT   EXEC FTG1CL,PARM.FORT='LOAD,NODECK,NOTERM,NOSOURCE'
C//FORT.SYSIN   DD  DSN=CLALQ01.CORRECT1.FORT,DISP=SHR
C//LKED.SYSLMOD  DD DSN=CLALQ01.INVEST.LOAD(INV),DISP=SHR,UNIT=DASD
C//LKED.SYSIN DD *
C  ENTRY MAIN
C  ALIAS ADIPIC,PND,LANDF
C NAME  INV(R)
C//
C
C     THE LIST BELOW IS THE JCL TO EXECUTE ON THE TVA SYSTEM
C     ALL LINES ARE IN COLUMN TWO HERE TO ALLOW FOR 'C' IN ONE
C
C//TXCOMECO JOB 123456,FSUDHOFF.CEBMS.2814,MSGLEVEL=1,CLASS=H,
C//            NOTIFY=CLALQ01
C//*MAIN  ORG=RGROUP03
C//PROCLIB DD DSN=L.SG508018.PROCLIB,DISP=SHR
C//*
```

```
C//SHAWNEE    EXEC  SCM,STPLIB=SHAW,PRTFMS=A
C//LOAD.DATA   DD  DSN=CLALQ01.TEST.DATA,DISP=SHR
C//INVEST.STEPLIB   DD    DISP=SHR,DSN=CLALQ01.INVEST.LOAD
C//REVENUE.STEPLIB   DD    DISP=SHR,DSN=CLALQ01.REV.LOAD(REV)
C//
C
C     THE LIST BELOW IS TO EXECUTE INTERACTIVE, SUBSTITUTE PND-
C     LANDF-ADIPIC ---INVEST.LOAD(ADIPIC)
C
C  FREEALL
C  FREE FI(FT03F001),FI(FT05F001),FI(FT06F001)
C  ALLOC FI(FT03F001) DA(*)
C  ALLOC FI(FT05F001) DA(*)
C  ALLOC FI(FT06F001) DA(*)
C  CALL INVEST.LOAD(ADIPIC)
C  FREEALL
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     MAIN DRIVER PROGRAM
C     REVISION DATE 82-02-23
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON /OPT/     BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
      COMMON/BIG1/BAGDLP,BAGRAT,BAGCOST,BGLIFE,PSSO2X,EBYPAS,RS,PSC,
     1 CCSA,EFFPS,ESPDLP,RESIST,SCARAT,FANHP,FANLC,FANMC,DELTAP,FILTSA,
     2 PSF,FILRAT,H,TL,VD,VE,TGASID,SKLAND,DRAINM,DRAINL,CLAYSK,WPSO3,
     3 WPSO4,WPCO3,WPINS,ASHCLN,WINDEX,PDMAX,FLOOD,SRAIR,OXAIR,OXSCFM,
     4 PRES,SO2RT,SO2RV,SO2RST,SDFEE,VPD,VT,HVCLN,HPTONW,
     5 ISLUDG,IFIXS,JSR,IVPD,IT,LON,LGMAX,ISTAND,NSBR,
     6 KEPASS,KPASO2,ICLAR,JSSO,JDIS,JSTR,JGPM,IRH,NSBF
      COMMON/TRAN/NOTRAN,NOREDN,NTOTAL,PCNTRN,NORAW,NOUTIL,NOUNIT,
     1            NOPROD,NOSHIP,NSPREP,NSCOMP,NOCOMP,NOBYDU
      COMMON/SCRUB/ISCRUB,IFOX,IADD,VLG,VTR,IFAN
      COMMON/ADDPRE/DUM1(15)
      COMMON/BOILER/DUM5(10)
      REAL MW
      COMMON/BYPAS2/DUM7(17)
      COMMON/BYPAS3/DUM8(14)
      COMMON/COAL/DUM12(11)
      COMMON/COMPLE/ICOMPL,IRPT,IRUN,NEXT,IEQPR,IWTBAL,ITM,INPOPT,NOPART
      REAL KGTA,KGEHT,KPCO2
      COMMON/CSTNDX/DUM14(8)
      COMMON/EMIT/DUM15(7)
      COMMON/EQCOST/DUM16(318)
      COMMON/EMIT/ISO2,XSO2,ISSO2,ELISO2,SO2REF,SO2LMB,SO2PPM
      COMMON/FILCON/DUM19(5)
      COMMON/FIXCOS/FIXADD,DS,TFIXEQ,TFIXLB,HPFIX(8)
      COMMON /PARM/ P(500)
      COMMON/FIXFDN/DUM23(5)
      REAL MFLOW(20,12),WFLOW(20,12),TOTFLO(20),GPM(20)
```

```
      COMMON/FLYASH/DUM25(7)
      COMMON/HGAS/ACFM,THG,YCO2,YHCL,YSO2,YO2,YN2,YH2O,ASH,WCL,WASH,IESP
      COMMON/HGAS2/DUM27(9)
      REAL MPCT(6),LBMH(6),LBH(6)
      COMMON/H2OAMT/DUM28(12)
      COMMON/IND/DUM29(16)
      COMMON/INPT/DUM30(5)
      COMMON/LANDIS/DUM35(7)
      DIMENSION SW(4),AREA(10),B(5),ULFC(7)
      REAL LGMAX
      COMMON/LIQDIS/DUM39(9)
      REAL MGPPM
      COMMON/LIQ2/DUM40(5)
      COMMON/LS/IALK,ISR,SR,SRADD,WPMGO,WPI,WPM,LSADD,LIMEAD,MGOADD,WMG
      REAL LSADD,LIMEAD,MGOADD
      COMMON/OUTPT/DUM42(5)
      COMMON/PCOST/DUM45(31)
      COMMON/PND1/DUM47(12)
      COMMON/PNDIND/DUM54(8)
      COMMON/SCRUBR/LG,V,PPMOUT,SO2R,ELIMIT,PR,PD,WPS,OX,PH,NSTAGE,
     &       NGRID,HS,ACS,ACSTRN
      COMMON/RCOAL/DUM57(9)
      COMMON/RRIND/DUM61(27)
      COMMON/RTIME/TR,ALCONT(9)
      REAL *8 XINT,PCTMNT,PDMNTP,XINFLA,PCTOVR,PCTADM,UNDCAP,PCTINS,UC(1
     1       8)
      COMMON/SCROXD/DUM62(8)
      COMMON/SCRUB/DUM63(6)
      COMMON/SCRUBC/CSCRUB(14),SCRUBL,SCRUBM
      COMMON/SCRUBR/DUM64(15)
      REAL LG
      COMMON/SKGAS/SSCFM,SACFM,TSK,SDUST,SWDUST,SMPCT,SLBMH,SLBH,TGAS
      REAL SMPCT(6),SLBMH(6),SLBH(6)
      COMMON/SOLDIS/DUM68(8)
      REAL INSOLS
      COMMON/SRHT/DUM69(19)
      DIMENSION ARH(6),DPRH(6),URH(6),NT(6),NB(6),OD(6),ARINCL(6),
     1          ARCORT(6),NINCLT(6),NCORTT(6),NINCLB(6),NCORTB(6)
      COMMON/TAXES/DUM70(8)
      COMMON/TRAN/DUM71(13)
      COMMON/WGAS/DUM72(8)
      REAL WMPCT(6),WLBMH(6),WLBH(6)
      COMMON/WGCOMP/DUM73(8)
      REAL NTRAIN
      INTEGER WING
      REAL *8 DSBCAP,DALAND,DWRCAP,DTOFLO,MGWATT,KBTU,DLBH3,
     1       DWLBH3,DWPS,AQ(18),DDIREC,DPOND
      DIMENSION DIREXC(9)
      LOGICAL KO,TVA
      INTEGER BOTTOM,CLNLVL,CT,FMB,TYPCT
      DIMENSION IAPCS(9),KO(2)
      COMMON /INPUT/    BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,
```

```
     &                PNAROM, ALKROM, CLROM, FEOROM, PMROM, HHVCLN, PSCLN,
     &                PACLM, CSTCLN, PNACLN, ALKCLN, CLCLN, FEOCLN, PMCLN,
     &                FMB, NR, IS, CT, TYPCT, CLNLVL, BOTTOM, NM, IAPCS, KO,
     &                TVA
      CHARACTER*4 INEXT
      CHARACTER*9 XXXALK(2)
      CHARACTER*28 XXSCRB(6)
      CHARACTER*11 XXIADD(3)
C
      DATA INEXT/'NEXT'/
      DATA XXXALK/'LIMESTONE','     LIME'/
      DATA XXSCRB/'SPRAY TOWER                  ','TURBULENT CONTACT
     &','VENTURI / SPRAY TOWER        ','VENTURI / SPRAY TOWER        ',
     &'VENTURI / TURBULENT CONTACT ','VENTURI / TURBULENT CONTACT '/
      DATA XXIADD/' NO         ',' MGO        ','ADIPIC ACID'/
C
      ICOMPL=0
      WING=1
      CALL ZERO
      DO 1 I=1,8
      HPFIX(I)=0.0
1     CONTINUE
      CALL READIN
C
      NEXT=0
      IF(ICOMPL.GT.1) GO TO 1030
1000  ICOMPL=0
C
      ISBCAL=1
C      CALL ZERO
C
      ISBCAL=2
C
      IF (ICOMPL.NE.0) GO TO 1010
C
      ISBCAL=3
      WING=2
      CALL BECHTL
C
      IF (ICOMPL.NE.0) GO TO 1010
C
      ISBCAL=4
C
      ISBCAL=5
      WING=3
      CALL WPASS
C      IF (ISLUDG.EQ.5) CALL SCLAND
C
      IF (IRPT.EQ.3) GO TO 1020
      IF (ICOMPL.NE.0) GO TO 1010
C
      ISBCAL=6
```

```
      WING=4
      CALL TVAIN(DIREXC)
C
 1010 CONTINUE
C
      ISBCAL=7
      WING=5
      CALL IAPPAS(DIREXC)
      WING=6
C
 1020 CONTINUE
 1030 CONTINUE
C
C
C     TVAFGD  MODULE-SPECIFIC  OUTPUT
C
C
C     SPECIFIC DESCRIPTION
C
      IF (KO(1)) WRITE(6,100) XXXALK(IALK),XXSCRB(ISCRUB)
      IF (KO(2)) WRITE(7,100) XXXALK(IALK),XXSCRB(ISCRUB)
  100 FORMAT(///T5,A9,' FGD',/T10,
     &'THE CONFIGURATION OF THIS SYSTEM INCLUDES ',A28,
     & ,/T9,' ABSORBERS. ')
C
      IF(IFOX.GE.1 .AND. IFOX.LE.3) THEN
        IF (KO(1)) WRITE(6,125)
        IF (KO(2)) WRITE(7,125)
      ENDIF
  125 FORMAT(1H+,T21,'FORCED OXIDATION IS USED TO STABILIZE THE ',
     &'SLURRY.')
C
      IF (KO(1)) WRITE(6,150) XXIADD(IADD+1)
      IF (KO(2)) WRITE(7,150) XXIADD(IADD+1)
  150 FORMAT(1H+,T70,A11,/T10,'CHEMICAL ADDITIVE IS USED.')
      IF (KO(1)) WRITE(6,200) NOREDN*100./NOTRAN,
     &LG,P(203),BYPFGD*100.,P(391)*100.
      IF (KO(2)) WRITE(7,200) NOREDN*100./NOTRAN,
     &LG,P(203),BYPFGD*100.,P(391)*100.
  200 FORMAT(T10,'SPARE ABSORBER CAPACITY OF',F4.0,'% IS PROVIDED.',
     &' THE L/G RATIO IS ',F5.1,/T10,'AND DESIGN SO2 REMOVAL OF',
     &F5.1,'% OCCURS IN THE TREATED GAS STREAM.'
     &/T10,F4.0,'% OF THE GAS STREAM IS BEING BYPASSED.'
     &/T10,F4.0,'% OF THE WASTES ARE DISPOSED OF IN AN ONSITE'
     &,' FACILITY.')
      RETURN
      END
```

```
      BLOCK DATA
      CHARACTER*4 HMOD,HSTRM,HCAP,SHCAP
      DIMENSION HMOD(10),HSTRM(5,8),NST(10),HCAP(10,10),SHCAP(6,8,10),
     &IC(8)
      COMMON /OUTH/ HMOD,HSTRM,HCAP,SHCAP,IC,NST
C
      DATA NST/0,0,0,0,2,2,3,7,0,0/
C
CC    OUTPUT HEADINGS
C
      DATA HMOD/3*'    ',' SH ','ESP ',' FF ','LSD ','LFGD','DSI ',
     &'WAST'/
      DATA HSTRM/' FLU','E GA','S, 1','000 ','LB/H',
     &          ' FLU','E GA','S, 1','000 ','ACFM',
     &          '   T','EMPE','RATU','RE, D','EG. F',
     &          '    ','   M','ISTU','RE, ','LB/H',
     &          '    ','ALKA','LINI','TY, ','LB/H',
     &          '  P','ARTI','CULA','TE, ','LB/H',
     &          '    ','    ','    ','  S','O2, ','LB/H',
     &          '    ','    ','    ','  N','O2, ','LB/H'/
      DATA HCAP/
     &'LIMB',9*'----',
     &'LOW ','NOX ','BURN','ERS/','OVER','FIRE',' AIR',13*'----',
     &'SPRA','Y HU','MIDI','FICA','TION',5*'----',
     &'ESP-',9*'----',
     &'FABR','IC F','ILTE','R---',6*'----',
     &'LIME','SPRA','Y DR','YING',6*'----',
     &'LIME','STON','E FG','D---',6*'----',
     &'DRY ','SORB','ENT ','INJE','CTIO','N---',4*'----',
     &'WAST','E DI','SPOS','AL ',6*'----'/
      DATA SHCAP/
     &192*'----',
     &'ESP&','DUCT','ING-',3*'----','ESP ','ASH ','DISP','OSAL',
     &38*'----',
     &'FF&D','UCTI','NG--',3*'----',
     &'FF A','SH D','ISPO','SAL-',2*'----',
     &36*'----',
     &'LSD ','REAG','ENT ','PREP',2*'----',
     &'LSD ','ABSO','RBER','S&DU','CTIN','G---',
     &'LSD ','RECY','CLE ','WAST','E DI','SP.-',30*'----',
     &'FGD ','MATE','RIAL',' HAN','DLIN','G---',
     &'FGD ','FEED',' PRE','PARA','TION','----',
     &'FGD ','GAS ','HAND','LING','----','----',
     &'FGD ','SO2 ','SCRU','BBIN','G---','----',
     &'FGD ','OXID','ATIO','N---','----','----',
     &'FGD ','REHE','AT--','----','----','----',
     &'FGD ','SOLI','D SE','PARA','TION','----',102*'----'/
      END
```

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM READIN
C     REVISION DATE 83-04-01
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE READIN
C
      COMMON /OPTIM/BYPSO2, BYPMST, BYPCFM
      COMMON /OPT/     BYPFGD, BYPFF, EFFESP, EFFLSD, EFFLIM
      COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
     1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
     2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
     3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
     4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
     5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
     6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
      COMMON /COAL1/ HHV, PS, PA, CST, PNA, ALK, CL, FEO, PM
      COMMON /STREAM/ PPH, ACFM1, TEMP, PPHMST, PPHALK, PPHPRT, PPHSO2, PPHNOX
      COMMON /INPUT/ BSIZE, CF, AHFX, AHXT, HHVROM, PSROM, PAROM, CSTROM, PNAROM,
     &ALKROM, CLROM, FEOROM, PMROM, HHVCLN, PSCLN, PACLN, CSTCLN, PNACLN, ALKCLN,
     &CLCLN, FEOCLN, PMCLN, FMB, NR, IS, CT, TYPCT, CLNLVL, BOTTOM, NM, IAPCS(9), KO
     &, TVA
      COMMON /INSYSP/ CCEFF, UTEFF, BNHR, FANLOS, QIN, ALKIN
      COMMON /PARM/ P(500)
      COMMON /MATBAL/ TABLE(8, 8)
      COMMON /NEW/ IFACT, VC, FC
      COMMON/ALKASH/ASHCAO, ASHMGO, SRCAO, SRMGO, SRTOT
      COMMON/BOILER/MW, BHR, EXSAIR, RMW, PREC, SPASH, WPRITE, KCLEAN, SMRW, SMCL
      REAL MW
      COMMON/BYPAS2/BYLBH(6), BYLBMH(6), BSCFM, BACFM, PASSBY, XPASSD, IBY
      COMMON/CID/CASEID(20)
      COMMON/COAL/HVC, WPC, WPH, WPO, WPN, WPSUL, WPCL, WPASH, WPH2O, SULO, ASHO
      COMMON/COMPLE/ICOMPL, IRPT, IRUN, NEXT, IEQPR, IWTBAL, ITM, INPOPT, NOPART
      COMMON/CONS/KGTA, KGEHT, KPCO2, C1, C2, C3, C4, PC1, PC2, PC3, PC4, PC5
      REAL KGTA, KGEHT, KPCO2
      COMMON/CSTNDX/ICEPYR, CEPNDX, INVYR, IREVYR, RATLAB, RATMAT, ARM, ARL
      COMMON/EMIT/ISO2, XSO2, ISSO2, ELISO2, SO2REF, SO2LMB, SO2PPM
      COMMON/FLOWS/GPM, MFLOW, WFLOW, TOTFLO
      REAL MFLOW(20, 12), WFLOW(20, 12), TOTFLO(20), GPM(20)
      COMMON/FLYASH/IASH, ASHUPS, ASHSCR, ASHPRU, ASHPRS, ASHLBU, ASHLBS
      COMMON/HGAS/ACFM, THG, YCO2, YHCL, YSO2, YO2, YN2, YH2O, ASH, WCL, WASH, IESP
      COMMON/HGAS2/MPCT, LBMH, LBH, SCFM, HHG, HWG, CFR, BCFR, RCFR
      REAL MPCT(6), LBMH(6), LBH(6), XCOMPX(8)
      COMMON/H2OAMT/SEEPRT, EVAPRT, RAIN, SEEP, EVAP, EVAPC, RAFAL, RAFALC,
     1              WLSM, H2OHYD, TWA, TWR
      COMMON/IND/ENGIN, ARCTEC, FLDEXP, FEES, CONT, START, CONINT, FILMAT(9)
      COMMON/INPT/JINPUT, JBC, JALK, JSSVAR,  JSRHT
      COMMON/JP/JP1, JP2, JP3, JP4, JP5, JP6, JP7, JP8, JP9, JP10
      COMMON/LANDIS/FILAND, TWLF, DISFUL, ELECLF, WATERF, FILANL, OVLF
```

```
      REAL LGMAX
      COMMON/LIQDIS/H2O, CAPPM, MGPPM, SO3PPM, SO4PPM, CLPPM, TDS, PHPOND, DGPM
      REAL MGPPM
      COMMON/LIQ2/IW4, AD, ADDC, ADLOSS, AMGLOS
      COMMON/LS/IALK, ISR, SR, SRADD, WPMGO, WPI, WPM, LSADD, LIMEAD, MGOADD, WMG
      REAL LSADD, LIMEAD, MGOADD
      COMMON/OPPSCH/IOPSCH, IYROP, IA(50)
      COMMON/OUTPT/JOUTPT, JIT, JEQ1, JEQ2, JEQ3
      COMMON/OUTPT1/JHGAS, JWGAS, JRAIR, JRGAS, JSRHO, JSKGAS
      COMMON/PCOST/POND(9), PDACRE, TOPHRS, PNDCAP, PNDLAB(9), PNDMAT(9), ELAN
      COMMON/PND1/PDEPTH, PMXEXC, DISTPD, ILINER, CLAYIN, CLAYUC, HYPLMC,
     1           HYPLLC, PSAMAX, ACREC, APRAIN, SAPILE
      COMMON/PNDIND/INDPND, PENGIN, PARCH, PFLDEX, PFEES, PCONT, PSTART, PCONIN
      COMMON/PT/ICONV, PHSTAR, CBSTAR, CB, CBR, HCO3EQ, SO3EQ, HSO3EQ
      COMMON/REHEET/OIL, OILGPM, HDUTY, RHSCFM, RDUST, RWDUST,
     &              RMPCT(6), RHLBMH(6), RHLBH(6)
      COMMON/RHAIR/ASCFM, ALBMH(6), ALBH(6)
      COMMON/RRIND/XINT, PCTMNT, PDMNTP, XINFLA, PCTOVR, PCTADM, UNDCAP,
     1             PCTINS, UC, IECON
      COMMON/RTIME/TR, ALCONT(9)
      REAL *8 XINT, PCTMNT, PDMNTP, XINFLA, PCTOVR, PCTADM, UNDCAP, PCTINS, UC(1
     1        8)
      COMMON/SCRUB/ISCRUB, IFOX, IADD, VLG, VTR, IFAN
      COMMON/SCRUBR/LG, V, PPMOUT, SO2R, ELIMIT, PR, PD, WPS, OX, PH, NSTAGE,
     1              NGRID, HS, ACS, ACSTRN
      REAL LG
      COMMON/SKGAS/SSCFM, SACFM, TSK, SDUST, SWDUST, SMPCT, SLBMH, SLBH, TGAS
      REAL SMPCT(6), SLBMH(6), SLBH(6)
      COMMON/SOLDIS/CASO3, CASO4, CACO3, INSOLS, WDIS, PSD, DENS, DEND
      REAL INSOLS
      COMMON/SRHT/TSTEAM, HVS, VRH, WSTEAM, ARH, DPRH, URH, Q, NT, NB, PL, OD,
     1            ARINCL, ARCORT, NINCLT, NCORTT, NINCLB, NCORTB, TEMPS
      DIMENSION ARH(6), DPRH(6), URH(6), NT(6), NB(6), OD(6), ARINCL(6),
     1          ARCORT(6), NINCLT(6), NCORTT(6), NINCLB(6), NCORTB(6)
      COMMON/TAXES/ITAXFR, TXFRRT, TXRAT, FRRAT, SERVRT, ROYALT, IOTIME, OTRATE
      COMMON/TRAN/NOTRAN, NOREDN, NTOTAL, PCNTRN, NORAW, NOUTIL, NOUNIT,
     1            NOPROD, NOSHIP, NSPREP, NSCOMP, NOCOMP, NOBYDU
      COMMON/WGAS/WACFM, WSCFM, TWG, ASHOUT, WASHOT, WMPCT, WLBMH, WLBH
      REAL WMPCT(6), WLBMH(6), WLBH(6)
      COMMON/WGCOMP/YWCO2, YWHCL, YWSO2, YWO2, YWN2, YWH2O, WVAPOR, NTRAIN
      REAL NTRAIN
      COMMON/WRIT9/DSBCAP, DALAND, DWRCAP, DTOFLO, MGWATT, KBTU, DLBH3,
     1             DWLBH3, DWPS, AQ, DDIREC, DPOND, IHVC
      REAL *8 DSBCAP, DALAND, DWRCAP, DTOFLO, MGWATT, KBTU, DLBH3,
     1        DWLBH3, DWPS, AQ(18), DDIREC, DPOND
      CHARACTER*4 IEND, INEXT, CASE, BLANK
      DIMENSION WCAQ(18)
C
      DATA IEND, INEXT/'END ', 'NEXT'/
      DATA CASE, BLANK/'CASE', '    '/
C
      BYPSO2=0.
```

```
BYPMST=0.
BYPCFM=0.
IECON= 1
ITAXFR= 1
EXSAIR=P(394)
SDFEE=P(162)
ACREC=P(181)
PSSO2X=P(203)
XSO2=P(203)
XLG=P(299)
XESP=P(301)
XRH=P(302)
KEPASS=P(303)
KPASO2=P(304)
TSK=P(305)
TSTEAM=P(306)
HVS=P(307)
IASH=P(308)
ASHUPS=P(309)
VLG=P(310)
VTR=P(311)
V=P(312)
VRH=P(313)
TR=P(314)
XIALK=P(315)
IADD=P(316)
WPMGO=P(317)
XMGOAD=P(318)
AD=P(319)
ADDC=P(320)
WPI=P(321)
SULO=P(5)*100.
ASHO=P(4)*100.
WPM=P(322)
WPS=P(323)
PSD=P(324)
RS=P(325)
PSC=P(326)
IFOX=P(327)
OX=P(328)
SRAIR=P(329)
PSF=P(330)
FILRAT=P(331)
PHLIME=P(332)
IVPD=P(333)
VPD=P(334)
DELTAP=P(335)
PRES=P(336)
IFAN=P(337)
ISCRUB=P(338)
XNS=P(339)
XNG=P(340)
```

```
      HS=P(341)
      RAIN=P(342)
      SEEPRT=P(343)
      EVAPRT=P(344)
      WINDEX=P(345)
      HPTONW=P(346)
      NOREDN=P(347)
      PCNTRN=P(348)
      ISLUDG=P(349)
      IFIXS=P(350)
      PSAMAX=P(351)
      PDEPTH=P(352)
      PMXEXC=P(353)
      DISTPD=P(354)
      ILINER=P(355)
      XLINA=P(356)
      IF(ILINER.EQ.2) XLINA=P(406)
      IF(ILINER.EQ.3) XLINA=0
      XLINB=P(357)
      IF(ILINER.EQ.2) XLINA=P(407)
      IF(ILINER.EQ.3) XLINA=0
      PNDCAP=P(391)
      NSPREP=P(392)
      NOTRAN=P(393)
C
CC    IF NUMBER OF OPERATING TRAINS IS 0 THEN DEFAULT BASED ON MW
C
      IF (NOTRAN .EQ. 0) THEN
        IF(BSIZE .LT. 100.) THEN
          NOTRAN= 1
        ELSE
          NOTRAN= INT(BSIZE)/250+2
          IF(NOTRAN .GT. 5) NOTRAN= 8
        ENDIF
      ENDIF
      ISO2=1
      ISR=P(300)
      SR=P(202)
      SRIN=SR
      IOPSCH=4
      ISUBSW=0
      TXRAT= P(374)
      FRRAT= P(375)
      TXFRRT=TXRAT+FRRAT
      SERVRT= P(376)
      MW=BSIZE
      BHR=BNHR
      ACFM=ACFM1*(1.-BYPFGD)
      BYPCFM=BYPFGD*ACFM1
      HVC=HHV
      THG=TEMP
      CV=.9*(VC-IFACT)
```

```
      IF (FC.EQ.0.0) THEN
         WPC=CV/.29
      ELSE
        WPC=CV+FC
       ENDIF
      IF(IFACT.EQ.14) WPH=VC*(7.35/(VC+10)-.013)-(.0125*PM)
      IF(IFACT.EQ.18) WPH=VC*(10.56/(VC+10)-.013)-(.0125*PM)
      IF (IFACT.EQ.14) WPN=2.10-.012*VC
      IF (IFACT.EQ.18) WPN=2.10-.046*VC
      WPSUL=PS
      WPCL=CL
      WPASH=PA
      WPH2O=PM
      WPO=100-(WPC+WPH+WPN+WPSUL+WPH2O+WPASH+WPCL)
      INPOPT=1
      CEPNDX=363.4
C
      IF (ISLUDG.GE.4) PSD=PSF
      IF (ILINER.EQ.0) GO TO 1050
      IF (ILINER-2) 1030,1040,1050
 1030 CLAYIN=XLINA
      CLAYUC=XLINB
      GO TO 1050
 1040 HYPLMC=XLINA
      HYPLLC=XLINB
 1050 CONTINUE
      IF(INDPND.EQ.1) GO TO 1060
      PENGIN=ENGIN
      PFLDEX=FLDEXP
      PFEES=FEES
      PCONT=CONT
      PSTART=START
 1060 CONTINUE
      PCONIN=CONINT
      NOPROD=1
      NORAW=4
      NOSHIP=0
      NOUTIL=4
      NOUNIT=11
      DO 1070 I=1,5
      IF (IOPSCH.EQ.I) GO TO 1080
 1070 CONTINUE
 2100 FORMAT('0****    INVALID OPERATING SCHEDULE CODE =',I10,
     1       5X,'CHANGED TO 1')
      IOPSCH=1
 1080 CONTINUE
      IYROP=30
C2300 FORMAT(A4)
 1110 CONTINUE
C
C      CALL EDIT
C
```

```
      IF(ICOMPL.EQ.1) RETURN
 2350 FORMAT(1H ,T11,20A4)
C     DETERMINE OPERATING HOURS
      GO TO (1120,1140,1200,1180,1160), IOPSCH
 1120 CONTINUE
C     FIVE PROCESS STUDY PROFILE
      DO 1130 I=1,IYROP
      IA(I)=7000
      IF (I.LE.10) GO TO 1130
      IA(I)=5000
      IF (I.LE.15) GO TO 1130
      IA(I)=3500
      IF (I.LE.20) GO TO 1130
      IA(I)=1500
 1130 CONTINUE
      GO TO 1200
 1140 CONTINUE
C     F.P.C. PROFILE
      DO 1150 I=1,IYROP
      IA(I)=5695
C     FROM THE EQUATION: IA(I)=0.65*8760.+1
      IF (I.LE.10) IA(I)=4381.0+131.4*I
C     FROM THE EQUATION: IA(I)=(50+1.5*I)/100.*8760.+1
      IF (I.GT.15) IA(I)=8060.2-157.68*I
C     FROM THE EQUATION: IA(I)=(92-1.8*I)/100.*8760.+1
 1150 CONTINUE
      GO TO 1200
 1160 CONTINUE
C     PROFILE FOR TVA/EPA PREMISES 12/79
      DO 1170 I =1,IYROP
      IA(I) = 6132
      IF(I.LE.5) IA(I) = 4381.0 + 350.4*I
      IF(I.GT.15) IA(I) = 10074.0 - 262.8*I
 1170 CONTINUE
      GO TO 1200
 1180 CONTINUE
      DO 1190 I = 1,IYROP
      IA(I) = 5500.
 1190 CONTINUE
 1200 CONTINUE
C     INPUT PROFILE
      TOPHRS=0.0
      DO 1210 I=1,IYROP
      IF (IA(I).GT.8784) IA(I)=IA(I)/10
      TOPHRS=TOPHRS+IA(I)
 1210 CONTINUE
      TOPHRS=TOPHRS*PNDCAP
      IF(IOTIME.EQ.1) RATLAB=RATLAB*(1.0+0.07*(OTRATE-1.0))
      ARM=1.0
      ARL=1.0
      CEPS = 0.01
      ICLAR = 0
```

```
      IF (PSC.GT.(WPS+CEPS)) ICLAR = 1
      LG = XLG
      VT=VPD
      NSTAGE = XNS
      NGRID = XNG
      IF (ISCRUB.EQ.2.OR.ISCRUB.EQ.5.OR.ISCRUB.EQ.6) GO TO 1220
      HS=0.0
      NSTAGE=0
 1220 CONTINUE
      JSR=ISR
      IALK = XIALK
      IF (IALK.NE.1.OR.IADD.EQ.2) PH=PHLIME
      MGOADD =XMGOAD
      JINPUT = XINPUT
      JOUTPT = OUTPUT
      JBC = XBC
      JALK = XALK
      JSSVAR = XSSV
      JSRHT = XSRHT
      JSSO = XSSO
      JDIS = XDIS
      JSTR = XSTR
      JGPM = XGPM
      JHGAS = XHGAS
      JWGAS = XWGAS
      JRAIR = XRAIR
      JRGAS = XRGAS
      JSRHO = XSRHO
      JSKGAS = XSKGAS
      JEQ1 = XEQ1
      JEQ2 = XEQ2
      JEQ3 = XEQ3
      JIT = XIT
      IESP = XESP
      IF (BYPFGD.NE.0) XRH=0
      IRH = XRH
      IF (JINPUT.NE.0) GO TO 1230
      JBC = 0
      JALK = 0
      JSSVAR = 0
      JSRHT = 0
 1230 CONTINUE
      IF (IRPT.NE.0) GO TO 1240
      JIT = 0
      JHGAS = 0
      JWGAS = 0
      JRAIR = 0
      JRGAS = 0
      JSRHO = 0
      JSKGAS = 0
      JEQ1 = 0
      JEQ2 = 0
```

```
      JEQ3 = 0
      JSSO = 0
      JDIS = 0
      JSTR = 0
      JGPM = 0
 1240 CONTINUE
      IF (IRH.EQ.2) JRAIR = 0
      IF (IRH.EQ.2) JRGAS = 0
      IF (ISR.NE.1) SR = SRIN
      IF (ISR.EQ.1) SR = 1.5
      IF (IALK.NE.1) SR = SRIN
      ISSO2 = ISO2
      ELISO2 = XSO2
      GO TO 1260
 1250 CONTINUE
 2400 FORMAT(///' **** FOUND END OF INPUT DATA BEFORE EXPECTED'/
     &     ' JOB TERMINATED.')
 1260 RETURN
      END
```

```
$DEBUG
$NOFLOATCALLS
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SHAWNEE LANDFILL  INPUT DATA
C
C     REVISION DATE 83-04-25
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ** *
C
      SUBROUTINE WASTE
C
      COMMON /COSTS/ CAPCST(10,8), ANN(25), INDCST(10)
      COMMON /PARM/P(500)
C
      COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
     1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
     2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
     3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
     4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
     5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
     6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
      COMMON/LACOS3/WTSM, WTSL, WTSL1, TLAB, CRATE, ZRATE, EQUIPC, NDOZER, NCOMP
      COMMON /INPUT/BSIZE, CF, AHFX, AHXT, HHVROM, PSROM, PAROM, CSTROM, PNAROM,
     &ALKROM, CLROM, FEOROM, PMROM, HHVCLN, PSCLN, PACLN, CSTCLN, PNACLN, ALKCLN,
     &CLCLN, FEOCLN, PMCLN, FMB, NR, IS, CT, TYPCT, CLNLVL, BOTTOM, NM, IAPCS(9), KO
     &, TVA
      COMMON/COAL/HVC, WPC, WPH, WPO, WPN, WPSUL, WPCL, WPASH, WPH2O, SULO, ASHO
      COMMON/CSTNDX/ICEPYR, CEPNDX, INVYR, IREVYR, RATLAB, RATMAT, ARM, ARL
      COMMON/FLYASH/IASH, ASHUPS, ASHSCR, ASHPRU, ASHPRS, ASHLBU, ASHLBS
      COMMON/HGAS2/MPCT, LBMH, LBH, SCFM, HHG, HWG, CFR, BCFR, RCFR
      COMMON/PCOST/POND(9), PDACRE, TOPHRS, PNDCAP, PNDLAB(9), PNDMAT(9), ELAN
      COMMON/PND1/PDEPTH, PMXEXC, DISTPD, ILINER, CLAYIN, CLAYUC, HYPLMC,
     1            HYPLLC, PSAMAX, ACREC, APRAIN, SAPILE
      COMMON/SCRUBR/LG, V, PPMOUT, SO2R, ELIMIT, PR, PD, WPS, OX, PH, NSTAGE,
     1              NGRID, HS, ACS, ACSTRN
      COMMON/SOLDIS/CASO3, CASO4, CACO3, INSOLS, WDIS, PSD, DENS, DEND
      COMMON/OPPSCH/IOPSCH, IYROP, IA(50)
      REAL MPCT(6), LBMH(6), LBH(6)
      COMMON/LANDIS/FILAND, TWLF, DISFUL, ELECLF, WATERF, FILANL, OVLF
      DIMENSION SW(4), AREA(10), B(5), ULFC(7)
      DIMENSION B1(5)
      REAL LG, IC(10), INDCST
      LOGICAL TVA, KO(2)
      REAL INSOLS
      COMMON/COMPLE/ICOMPL, IRPT, IRUN, NEXT, IEQPR, IWTBAL, ITM, INPOPT, NOPART
      COMMON/PCOSTS/PCLEAR, PDEXEC, SODM, SODL, PERIML, PERIMM, ROADMM, ROADLL,
     1 RECLAM
      COMMON/PNDARW/AB, AID, AIE, AOD, AS, DL, VEC, VETT, WELLM, WELLL, WPSITE
      COMMON/PNDIND/INDPND, PENGIN, PARCH, PFLDEX, PFEES, PCONT, PSTART, PCONIN
      COMMON/RRIND/XINT, PCTMNT, PDMNTP, XINFLA, PCTOVR, PCTADM, UNDCAP,
```

```
     1          PCTINS,UC,IECON
      REAL *8 XINT,PCTMNT,PDMNTP,XINFLA,PCTOVR,PCTADM,UNDCAP,PCTINS,UC(1
     1        8)
      DIMENSION CYD(7),TYD(7),FUELDT(7),BYD(5)
      COMMON/TAXES/ITAXFR,TXFRRT,TXRAT,FRRAT,SERVRT,ROYALT,IOTIME,OTRATE
      DATA CYD/28700.,30500.,42500.,44100.,46545.,47500.,71050./
      DATA TYD/3.,6.,12.,14.,16.,18.,26./
      DATA FUELDT/9.,8.,7.5,7.0,6.5,6.5,5.0/
      DATA BYD/1.5,2.25,3.,5.25,7./
C     DENS, DEND--- DENSITY SITE, DISPOSAL
C     A1 - ANGLE, HLIFT - LIFT HEIGHT, CAP - CLAY COVER, CAPS-SYNTHIC
C     SW(3) OFFSET WIDTH, YD3 SITE YD3, XLT - DISPOSAL YD3
C     ULFC  1 - SYNTHETIC MAT, 2 - SYNTHETIC LABOR, 3 - 8 IN DRAIN
C     4 - 4 IN DRAIN, 5 REDUCE TEES, 6 GRAVEL MAT, 7- GRAVEL LABOR
      TAXFR= P(228)
      WRITE(8,*) CLAYUC,TXFRRT,RATLAB
      LD= 8
      ITAXFR=1
      IRPT=1
      WRITE(8,*) 'DEND ='
      DEND = 70.
      IF(OX.GE.95.) DEND = 80.
      IF(PDEPTH.GT.30.) DEND = PDEPTH
      DENS = 85.
      IF(OX.GE.95.) DENS = 95.
      IF(PMXEXC.GT.30.) DENS = PMXEXC
      WOTH=ANN(13)*2000/(8760.*CF)
      ANN(12)= ANN(12)+WDIS/2000.*8760.*CF
      IF (PNDCAP .EQ. 0.0) GO TO 888
      YD3 =( WDIS+WOTH )/ (DENS * 27)
      YD3 = YD3 * TOPHRS * PNDCAP
      XLT = ( (WDIS+WOTH)*PNDCAP) / (DEND * 27.)
      IF (IFIXS.LT.1) GO TO 1000
      DEND = 90.
      IF(OX.LT.50.) DEND = 95.
      IF(PDEPTH.GT.30.) DEND = PDEPTH
      DENS = 100.
      IF(OX.LT.50.) DENS = 106.
      IF(PMXEXC.GT.30.) DENS = PMXEXC
      FLIASH = WOTH
      DRYSLD = WDIS * PSD/100.
      FIXADD = 0.035 * (FLIASH + DRYSLD)
      PMIXER = FLIASH +DRYSLD + FIXADD
      IF(PNDCAP.NE.1.0) GO TO 1000
      PNDCAP = PMIXER / (WDIS+WOTH)
      WRITE(8,*) ' PNDCAP='
      YD3 = ((WDIS+WOTH))/(DENS*27.)
      YD3 = YD3*TOPHRS*PNDCAP
      XLT = ((WDIS+WOTH)*PNDCAP)/(DEND*27.)
1000  CONTINUE
      A1 = 6
      HLIFT = 20
```

```
      CAP = 3.08
      CAPS = 4.80
      SW(3) = 10
      ULFC(6) = .28
      ULFC(7) =.23
      OPP = IA(1)
      ARM = RATMAT
      ARL = RATLAB
      STORM = 8.0 / 12.0
      ULFC(1) = HYPLMC / 9.
      ULFC(2) = HYPLLC / 9.
      ULFC(3) = 4.73
      ULFC(4) = 2.06
      ULFC(5) = 12.03
      IF(ILINER.NE.1) ULFC(3) = 0.0
      IF(ILINER.NE.1) ULFC(4) = 0.0
      IF(ILINER.NE.1) ULFC(5) = 0.0
      IF(ILINER.EQ.1) CAP = CLAYUC
      WRITE(8,*) ' GRAVEL ='
C
C     OFFSET AREA          1
C     SLOPE AREA           2
C     CAP AREA             3
C     LANDFILL SITE AREA   4
C     PILE BOTTOM SLAREA   5
C     POND AREA            6
C     CLAY AREA            7
C     PERIMETER FENCE      8
C     POND BOTTOM         10
C
      GRAVEL = 0.0
      DMIN = 80
      DMAX = 8000
      DTOE = 2500
      PI = 3.141592654
      A  = (A1*2*PI)/360
      SW(1) = 2 * HLIFT
      SW(2) = SW(1) * 2
      SW(4) = SW(3) * 2
      FAC = 1. /  3.
      XLF = YD3
C     INPUT YD3 CONVERTED TO FT3  FOR LFILL CONSTRUCTION
C     DETERMIND LANDFILL DESIGN
C     DTOE IS LENGTH OF SLUDGE DISPOSAL SITE
      YD3 = YD3 * 27
      DO 100 I = 1,20
      B1(1) = DTOE **2
      B1(2) = (DTOE - SW(2))**2
      CL = (DTOE-SW(2)-SW(4))
      B1(3) = CL**2
      ALT = TAN(A) * (CL/2)
      VOLC1 = (FAC*HLIFT)*(B1(1)+B1(2)+(B1(1)*B1(2))**.5)
```

```
      VOLC2 = FAC*ALT*B1(3)
      VOLC = VOLC1 + VOLC2
      VOL = YD3 - VOLC
      IF(VOL.GT.0) GO TO 10
      IF(VOL.LE.0) GO TO 20
   10 DMIN = DTOE
      DTOE = (DTOE+DMAX)/2
      GO TO 200
   20 DMAX = DTOE
      DTOE = (DTOE + DMIN)/2
  200 CONTINUE
  100 CONTINUE
      CENTRO =((.5*HLIFT*VOLC1) +(FAC*ALT*VOLC2))/VOLC
C     TAN 5 DEGREES = 0.08749, LARGEST SLOPE TRUCKS- AT 5 MPH
      CENT = 6 * CENTRO / 0.08749
      B1(1) = DTOE **2
      B1(2) = (DTOE - SW(2))**2
      CL = (DTOE-SW(2)-SW(4))
      B1(3) = CL**2
      B1(4) =  SQRT(B1(3))
      B1(5) = B1(4)/2
      AREA(1) = B1(2) - B1(3)
      AREA(2) = 4*(.5*(DTOE+DTOE-SW(2))*(SQRT(HLIFT**2 + SW(1)**2)))
      AREA(3) = 4*.5*B1(4)*SQRT(B1(5)**2 + ALT**2)
      AREA(4) = AREA(1) + AREA(2) + AREA(3)
      CATCH = 900
      DEP   = 20 + (DTOE - 80) * 0.0025
      PMAX = 1250
      PMIN = 4 * DEP
      PTOE = 400
C     DETERMIND CATCHMENT BASIN DESIGN
C     DEPTH VARIES BY LENTH OF DTOE
C     PTOE IS BASIN SIDE LENGTH
C     RAINSA IS INPUT STORM RAIN ON DISPOSAL SITE + BASIN + TRENCH
C     AREA + 10 % FREEBOARD
      DO 500 KI =1,10
      DO 600 IK = 1,20
      RAINSA = (DTOE + 24) **2 + CATCH
      VOLPC = RAINSA * STORM * 1.1
      CB1 = PTOE **2
      CB2 = (PTOE - (2 * 2 * DEP)) **2
      VOLP = (.333333 * DEP) * ((CB1 + CB2) + SQRT(CB1 * CB2))
      VOLCB = VOLPC - VOLP
      IF(VOLCB.GT.0) GO TO 30
      IF(VOLCB.LE.0) GO TO 40
   30 PMIN = PTOE
      PTOE = (PTOE + PMAX) / 2
      GO TO 700
   40 PMAX = PTOE
      PTOE = (PTOE + PMIN) / 2
  700 CONTINUE
  600 CONTINUE
```

```
      CATCHN = PTOE * PTOE
      BASE = CATCHN - CATCH
      BASE = ABS(BASE)
      CATCH = CATCHN
      IF(BASE.LT.50) GO TO 900
  500 CONTINUE
  900 CONTINUE
      CB2 = (PTOE - (2 * 2 * DEP)) ** 2
      CB1 = PTOE **2
      VOLP = (.333333 * DEP) * ((CB1 + CB2) + SQRT(CB1*CB2))
C     TRENCH INCLUDES 50 FEET DISPOSAL TO CATCHMENT BASIN
      DTRECH =(4 * (DTOE + 24)) + 50
      PS = DEP*2
      AREA(5) = DTOE **2
      AREA(6) = 4 * .5 * (PTOE + PTOE - PS) * SQRT(DEP**2 + PS**2)
      AREA(7) = DTRECH * 25.298 + AREA(6) + AREA(5)
      WTOE = PTOE
      AREA(10) = (PTOE - DEP) ** 2
C     LINER AREA SUM OF WETTED TRENCH + BASIN + LFILL DISPOSAL BOTTOM
C     GRAVEL-ALL WETTED TRENCH AND BASIN SLOPE, ALSO COVERS DRAIN PIPE
      TLINER = AREA(7)
      IF(ILINER.LT.2) GRAVEL = (25.298 * DTRECH + AREA(6)) / 2
      VLF = AREA(5) * 1.5
      VTRENH =(( .5 * 24 * 4) + 1.5) * DTRECH
C     EXCAVATION INCLUDES 1.5 FEET TOPSOIL OVER ENTIRE DISPOSAL +
C     ENTIRE BASIN + TRENCH
      EXCYD3 = (VTRENH + (VLF) + VOLP) / 27
      TH = HLIFT + ALT
C     DRAIN PIPE AT 100 FEET APART UNDER DISPOSAL, 4 IN CONNECTS TO
C     8 IN COLLECTORS
      DRAIN8 = 4 * DTOE
      DRAINN = (DTOE-100)/100
      DRAIN4 = DRAINN * DTOE
      TEES = 4 * DRAINN
      PEXC = (DRAIN8 + DRAIN4) * 1 * 2
      IF(ILINER.NE.1) PEXC = 0.0
      DRAINT = DRAIN8 * ULFC(3) + DRAIN4 * ULFC(4) + TEES * ULFC(5)
      IF(DRAINT.EQ.0) PEXC = 0.0
      GRAVEL = GRAVEL + PEXC
      VDTENC = 4*(FAC*((DTOE+24)*4*4)+(4*2*(DTOE+24)))
      VSTREN = VTRENH - VDTENC
      VOLPD = 4*(FAC*((PTOE*6*6)+(6*2*PTOE)))
      VSP = VOLP - VOLPD
      VPILE = VSP + VSTREN + VLF + PEXC
C     TOPSOIL PILE
      WTPMIN=12
      DPILE=10.0
      HPILE=20.0
      XLPILE = DTOE - PTOE -12
 1010 CONTINUE
C     WIDTH OF STORAGE PILE
      WPILE=(VPILE/HPILE+3.0*XLPILE*HPILE-12.0*HPILE*HPILE)/
```

```
     1       (XLPILE-3.0*HPILE)
        WTPILE=WPILE-6.0*HPILE
        IF(WTPILE.GT.WTPMIN) GO TO 1020
 2000 FORMAT(//10X,'****     TOPSOIL STORAGE PILE DESIGN ERROR     ****'
     1         /10X,'                    RUN TERMINATED')
        IF(HPILE.LT.1.5) STOP
        HPILE=HPILE-1.0
        GO TO 1010
C
 1020 CONTINUE
C       SURFACE AREA OF PILE
        SAPILE=XLPILE*WPILE+6.0*HPILE*(SQRT(10.)/3.0-1.0)*
     1       (XLPILE+WPILE-6.0*HPILE)
C       LAND AREA FOR STORAGE PILE
        APILE=XLPILE*(WPILE+DPILE)
C       SEED AREA
        SAREA = DTOE + 24 + 2
        SURFAE = ((52 + SAREA)**2 - (SAREA**2))
C       LANDFILL DIKE
        DL = 16+4
        DIKELF = (4 *(4 + SAREA)* SQRT(DL)) + 4 * 2 * SAREA
C       BASIN DIKE
        PL = 36+9
        DIKEP  = (4 *(6 + PTOE) *  SQRT(PL)) + 4 * 2 * PTOE
C       STRIP BETWEEN LANDFILL DISPOSAL AND STORAGE OR BASIN
        SURPAT = SAREA * 24
        SURE = ABS(PTOE-WPILE)
        SUREXT = SAREA * SURE
        CORNER = 64 + 4 * ((52*50) + 116)
        IF(PTOE.GT.WPILE) SPOND = WTOE * (SAREA - PTOE)
        IF(PTOE.LE.WPILE) SPOND = WTOE * (SAREA - WPILE)
        SEED =( SURFAE+DIKELF+DIKEP+SURPAT+SUREXT+SAPILE+CORNER+SPOND)/9
C       CHECKING IF POND OR STORAGE IS LARGER
        IF(WPILE.GT.PTOE) WTOE = WPILE
        FENCEL = (DTOE + 24 + 50 + 24 + 10 + 6 + 50 + 24 + 6)
C        3-3-3,50,TRENCH,DIKE,,LF-TRENCH-DIKE-PDIKE-ROAD-POND-DIKE-50
        FENCEW = (DTOE + 24 + 6 + 8 + 24 + WTOE + 8 + 50+ 50+ 6)
C       FENCE WILL ENCLOSE ENTIRE BASIN AND ENTIRE LANDFILL SITE
        AREA(8) =( FENCEL + FENCEW + PTOE) * 2
        AREA(9) = FENCEL * FENCEW / 43560
        ROAD = 2 * (FENCEL + WTOE)
C       CONVERT BACK TO YD3
        VPILE = VPILE / 27
        YD3 = XLF
        RCL = 0.0
        RCM = 0.0
        RCT = 0.0
        TIME1 = 1.5
        SHIFT = 2080 * 2
        II = 0
        COSTB = 10000000.0
        DRATE = TIME1 * XLT
```

```
C       CYCLE TIME IS 20 MINUTES LOAD-UNLOAD + DISTANCE TRAVEL AT 20 MPH
        DISTAN = (DISTPD * 2 + DTOE + CENT)/5280
        CYCLE = .33 + DISTAN/20
        II= 7
        DO 31 I =1,7
        TRIP = DRATE/TYD(I)
        NTRUCK = (TRIP * CYCLE) + 1
        DISFUL = DISTAN/FUELDT(I) * SHIFT/CYCLE
C       OPTIMIZE ASSUME 8 YEAR TRUCK LIFE
        COST =(NTRUCK+1)*(CYD(I)/8+.3*CYD(I)) + NTRUCK * (SHIFT*UC(6)+
       1DISFUL*UC(10))
        IF(COST.GT.COSTB) GO TO 25
        COSTB = COST
        ITRUCK = NTRUCK
        II = I
   25 CONTINUE
   31 CONTINUE
        COSTB = (ITRUCK + 1) * CYD(II) * RATMAT
C       DIESEL FUEL FOR TRUCKS ONLY
        DISFUL = DISTAN/FUELDT(II) * ITRUCK * SHIFT/CYCLE
        TWLF = ITRUCK * SHIFT
C       COSTB IS FOR TRUCKS,
C       WHEEL LOADERS
        NLOAD = 0
   11 NLOAD = NLOAD + 1
        DO 15 K =1,5
        TOPASS = DRATE * (4/.8)/(BYD(K)*NLOAD)
        PTIME = 60./TOPASS
        IF(PTIME.GT.1.0) GO TO 21
   15 CONTINUE
        GO TO 11
   21 FUELL = NLOAD * SHIFT * 2.0224 * EXP(.2701*BYD(K))
        TOLLAB = SHIFT * NLOAD
        COSTL = NLOAD * RATMAT * 45558.3*EXP(.292462*BYD(K))
        CUPYD3 = BYD(K) / 2
C       CLEAN-UP LOADER
        FUELUP = SHIFT * 2.0224 * EXP(.2701*CUPYD3)
        COSTCU = RATMAT * 45558.3 * EXP(.292462 * CUPYD3)
        TOCLAB = SHIFT
C       TRACK DOZER
        ZRATE = DRATE
        NDOZER = 1
        IF(DRATE.LT.120.) GO TO 26
        NDOZER = 2
        ZRATE = DRATE/2
   26 CONTINUE
        FUELD = SHIFT * NDOZER * 2.1305 * EXP(.010369 * ZRATE)
        COSTD = (RATMAT) * (39584.8 + 1481.37 * ZRATE)* NDOZER
        DHP = 45.0404 +(ZRATE * 1.27726)
        TODLAB = SHIFT * NDOZER
C       COMPACTOR
        NCOMP = 1
```

```
      CRATE = DRATE
      IF(DRATE.LT.105.) GO TO 27
      NCOMP = 2
      CRATE = DRATE / 2
   27 CONTINUE
      FUELC = 1.47179 * NCOMP *  CRATE**.63471
      COSTC = RATMAT * NCOMP *69565.4*EXP(CRATE*0.0138085)
      TACLAB = SHIFT * NCOMP
C     WATER TRUCK
      FUELT = SHIFT * 0.1
      COSTT = 34800. * RATMAT
      TOTLAB = 0.5 * SHIFT
C     SERVICE TRUCK
      DRATE2 = DRATE/3
      FUELS = SHIFT * 0.1
      COSTS = RATMAT*4434.89*DRATE2**.774537
      TOSLAB = 0.5 * SHIFT
C     TRAILER
      TRAIL = 10000.
      IX = 12
      IL = 30
      EL = 10000.
      IF(DISTPD.LT.20000) GO TO 3900
      EL = 15500.
      IX = 14
      IL = 40
      TRAIL = 15000.
 3900 CONTINUE
      TLAB = TRAIL*0.1*RATLAB
      TRAIL = TRAIL*RATMAT
C     WATER TREATMENT SYSTEM
      WTSM = 29500. * RATMAT * (XLT/44.2)**.6
      WTSL = 22500. * RATLAB * (XLT/44.2)**.6
      ELP = 60000. * (XLT/44.2)**.6
C     TOTALS  - FUEL, LABOR, EQUIPMENT
      DISFUL = DISFUL+FUELL+FUELUP+FUELD+FUELT+FUELS+FUELC
      TWLF = TWLF+TOLLAB+TOCLAB+TODLAB+TACLAB+TOTLAB+TOSLAB
      EQUIPC = COSTB+COSTL+COSTCU+COSTD+COSTC+COSTT+COSTS+TRAIL+FWTSM
      TYD3 = YD3 / TOPHRS
      IF(IRPT.NE.0) WRITE(LD,2050)
 2050 FORMAT(1H1)
      WRITE(LD,2150)
 2150 FORMAT(T30,'LANDFILL DESIGN'/T30,'-------- ------'/)
      IF (PNDCAP.NE.1.0) WRITE(LD,2350) PNDCAP
 2350 FORMAT(/T5,'(LANDFILL DESIGNED FOR',2PF7.2,' % OF PROJECTED ',
     1       'LIFETIME CAPACITY)'/)
      IF(IFIXS.GT.0.AND.PNDCAP.NE.1.0) WRITE(LD,2375)
 2375 FORMAT(T5,'(LANDFILL VOLUME INCLUDES SLUDGE FIXATION PROCESS',
     1' VOLUME)',/)
      SITE = AREA(9) * 43560
      ALFILL = AREA(4)
      RAINSA = RAINSA
```

```
      ACREFT=27 * YD3/43560.0
      IF (IRPT.EQ.0) GO TO 1030
      WRITE(LD,2400)
 2400 FORMAT(/T8,'LANDFILL DIMENSIONS'/T8,'-------- ----------'/)
      WRITE(LD,2450) TH,ALT,A1,DTOE,DTRECH,AREA(8)
 2450 FORMAT(T8,'HEIGHT OF LANDFILL',T40,F9.2,T50,'FT'/
     1       T8,'HEIGHT OF LANDFILL CAP',T40,F9.2,T50,'FT'/
     2       T8,'SLOPE OF LANDFILL CAP',T40,F7.0,T50,'DEGREES'/
     3       T8,'LENGTH OF LANDFILL DISPOSAL SIDE ',T40,F7.0,T50,'FT'/
     4       T8,'LENGTH OF LANDFILL TRENCH',T40,F7.0,T50,'FT'/
     5       T8,'LENGTH OF PERIMETER FENCE',T40,F7.0,T50,'FT'/)
      WRITE(LD,2500) ALFILL,RAINSA,SAPILE,AREA(5),SITE,AREA(9)
 2500 FORMAT(T8,'SURFACE AREA OF LANDFILL',T40,-3PF7.0,T50,'THOUSAND FT2
     1'/T8,'FILL AREA LAND EXPOSED TO RAIN',T40,F7.0,T50,'THOUSAND FT2'
     2/T8,'SURFACE AREA OF RECLAIM STORAGE',T40,F7.0,T50,'THOUSAND FT2'/
     3 T8,'DISPOSAL LAND AREA OF LANDFILL',T40,F7.0,T50,'THOUSAND FT2'/
     4 T8,'LAND AREA OF LANDFILL SITE',T40,F7.0,T50,'THOUSAND FT2'/
     5 T8,'LAND AREA OF LANDFILL SITE',T40,0PF7.0,T50,'ACRES'/)
      WRITE(LD,2550) EXCYD3,VPILE
 2550 FORMAT(T8,'VOLUME OF EXCAVATION',T40,-3PF7.0,T50,'THOUSAND YD3'/
     1       T8,'VOLUME OF RECLAIM STORAGE',T40,-3PF7.0,T50,
     1       'THOUSAND YD3')
      WRITE(LD,2600) YD3,ACREFT
 2600 FORMAT(T8,'VOLUME OF SLUDGE TO BE',T40,-3PF7.0,T50,'THOUSAND YD3'
     2 /T9,'DISPOSED OVER LIFE OF PLANT',T40,0PF7.0,T50,'ACRE FT'/)
      IF(IFIXS.LT.1) GO TO 2602
      FLYYD3 = (FLIASH * TOPHRS)/(DENS*27)
      FLYACR = (27. * FLYYD3)/43560.
      WRITE(LD,2601) FLYYD3,FLYACR
 2601 FORMAT(T8,'VOLUME OF FLYASH TO BE',T40,-3PF7.0,T50,'THOUSAND',
     2' YD3'/T9,'DISPOSED OVER LIFE OF PLANT',T40,0PF7.0,T50,'ACRE FT'/)
 2602 CONTINUE
      WRITE(LD,2451) DEND,DENS
 2451 FORMAT(T8,'DENSITY OF DISCHARGE CAKE',T40,F9.2,T50,
     1       'LBS/FT3',/T8,'DENSITY OF COMPACTED CAKE',T40,F9.2,
     2       T50,'LBS/FT3',/)
 1030 CONTINUE
      VOLPP = VOLP / 27
      IF(IRPT.NE.0) WRITE(LD,2452) DEP,PTOE,VOLPP
 2452 FORMAT(T8,'DEPTH OF CATCHMENT POND',T40,F9.2,T50,'FT'/
     1       T8,'LENGTH OF CATCHMENT POND',T40,F9.2,T50,'FT'/
     2   T8,'VOLUME OF CATCHMENT POND',T40,-3PF7.0,T50,'THOUSAND YD3'//)
      WTSL1 = WTSL
      WTSL = WTSL + TLAB
      CDM =  EQUIPC
      CDM2 = EQUIPC+WTSL
      WRITE(LD,2650)
 2650 FORMAT(T8,'LANDFILL COSTS (THOUSANDS OF DOLLARS)'/
     1       T8,'-------- ----- ---------- -- --------'//)
      IF(ITAXFR.EQ.1) WRITE(LD,2651) CDM2
 2651 FORMAT(T8,'LANDFILL EQUIPMENT',T60,-3PF8.0)
      EFTAX = 0.0
```

```
      IF(ITAXFR.EQ.1) EFTAX = CDM * TXFRRT/100.
      CDET = CDM + EFTAX + WTSL
      IF(ITAXFR.EQ.1) WRITE(LD,2660) EFTAX
 2660 FORMAT(T8,'TAX AND FREIGHT',T60,-3PF8.0/)
      WRITE(LD,2661) CDET
 2661 FORMAT(T8,'LANDFILL EQUIPMENT TOTAL',T60,-3PF8.0/T8,60('-')/
     2       T43,'LABOR',T52,'MATERIAL',T63,'TOTAL'/
     3       T43,'-----',T52,'---------',T63,'-----'/)
C     ARM = RATMAT IF SCM, OR 1 IF TERMINAL
C     ARL = RATLAB IF SCM OR 1 IF TERMINAL-- INCLUDES OVERTIME IF
C     SPECIFIED BY INPUTS
      CCL = AREA(9) * PCLEAR * ARL
      CEL = EXCYD3 * PDEXEC  * ARL
80000 FORMAT (' CEL = ',F10.3,'  EXCYD3 = ',F10.3,'  PDEXEC = ',F10.3,
     &'  ARL = ',F10.3)
      CGM = GRAVEL * ULFC(6) * ARM
      CGL = GRAVEL * ULFC(7) * ARL
      IF(ILINER.EQ.0) AREA(7) = 0.0
      CLCL = AREA(7) * ARL* CLAYIN * CLAYUC /(12.*27.)
      CLHL = AREA(7) * ULFC(2) * ARL
      CLHM = AREA(7) * ULFC(1) * ARM
      LABBOO = 0
      CDL = LABBOO * RATLAB
      CDDM = DRAINT * 0.91 * ARM
      CDDL = DRAINT * 0.09 * ARL
      CSL = SEED * SODL * ARL
      CSM = SEED * SODM * ARM
      CRL = ROAD * ROADLL * ARL
      CRM = ROAD * ROADMM * ARM
C     PROAD IS LAND REQUIRED FROM PLANT TO SITE
C     SROAD IS 24 FOOT WIDE ROAD FROM PLANT TO SITE
      PROAD = 44.* DISTPD / 43560.
      SROAD = (24.* DISTPD) / (2.* 27.)
      CRL = CRL + (PROAD*PCLEAR*ARL) + (ROADLL*ARL*SROAD)
      CRM = CRM + ROADMM * SROAD * ARM
      CPPL = AREA(8) * PERIML * ARL
      CPPM = AREA(8) * PERIMM * ARM
      CA = ACREC * AREA(9)
      CLHT=CLHL+CLHM
      CGT = CGL + CGM
      CST = CSL + CSM
      CRT=CRL+CRM
      CDT = CDL + CDM
      CDDT = CDDL + CDDM
      CPPT=CPPL+CPPM
      RECLAM = RECLAM * ARL * AREA(4)/43560.
80020 FORMAT (' RECLAM = ',F15.3,'  ARL = ',F10.3,'  AREA(4) = ',F15.3)
      WELLM = WELLM * ARM
      WELLL = WELLL * ARL
C     DISCHARGE TRENCH AT $ 39.00 FOOT, 30% OF DTOE LENGTH
      DICHAR = .3 * DTOE * 39.00 * RATLAB
      IF(ILINER.NE.1) GO TO 102
```

```
      RCL = ARL * ALFILL * CAP * 6./(12.* 27.)
      RCM = 0.0
  102 CONTINUE
      IF(ILINER.NE.2) GO TO 201
      RCL =(((CAPS/9) * .25)+(CAP*6/(12*27))) * ARL * ALFILL
      RCM = (CAPS/9) * .75 * ARM * ALFILL
      RCT = RCL + RCM
  201 CONTINUE
      WELLT = WELLM + WELLL
      IF(IRPT.EQ.0) GO TO 1040
      WRITE(LD,2700) CCL,CCL,CEL,CEL,DICHAR,DICHAR
 2700 FORMAT(T8,'CLEARING LAND',T40,-3PF8.0,T60,F8.0/
     1       T8,'EXCAVATION',T40,F8.0,T60,F8.0/
     3       T8,'DISCHARGE TRENCH',T40,F8.0,T60,F8.0)
      IF(ILINER.NE.2) WRITE(LD,2710) CGL,CGM,CGT
 2710 FORMAT(T8,'GRAVEL',T40,-3PF8.0,T50,F8.0,T60,F8.0)
      IF (ILINER.EQ.1) WRITE(LD,2750) CLAYIN,CLCL,CLCL
 2750 FORMAT(T8,'LINING(',F4.0,' IN. CLAY)',T40,-3PF8.0,T60,F8.0)
      IF (ILINER.EQ.2) WRITE(LD,2800) CLHL,CLHM,CLHT
 2800 FORMAT(T8,'LINING(SYNTHETIC)',T40,-3PF8.0,T50,F8.0,T60,F8.0)
      WRITE(LD,285)CDDL,CDDM,CDDT,CSL,CSM,CST,CRL,CRM,CRT,CPPL,CPPM,CPPT
  285 FORMAT(T8,'DRAINAGE LANDFILL ',T40,-3PF8.0,T50,F8.0,T60,F8.0/
     1       T8,'SEEDING LANDFILL SITE ',T40,-3PF8.0,T50,F8.0,T60,F8.0/
     1       T8,'ROAD CONSTRUCTION',T40,F8.0,T50,F8.0,T60,F8.0/
     2       T8,'PERIMETER COSTS, FENCE',T40,F8.0,T50,F8.0,T60,F8.0)
      WRITE(LD,2900) RECLAM,RECLAM
 2900 FORMAT(T8,'RECLAMATION EXPENSE',T40,-3PF8.0,T60,F8.0)
      IF(ILINER.EQ.1) WRITE(LD,3300) RCL,RCL
 3300 FORMAT(T8,'RECLAMATION CLAY COVER',T40,-3PF8.0,T60,F8.0)
      IF(ILINER.EQ.2) WRITE(LD,3400) RCL,RCM,RCT
 3400 FORMAT(T8,'RECLAMATION SYNTHETIC COVER',T40,-3PF8.0,
     1       T50,F8.0,T60,F8.0)
      WRITE(LD,3500) WELLL,WELLM,WELLT
 3500 FORMAT(T8,'MONITOR WELLS',T40,-3PF8.0,T50,F8.0,T60,F8.0/
     1       T8,60('-'))
 1040 CONTINUE
C     IF PROGRAM MODIFICATIONS MADE INCLUDE ADDITIONS HERE

      CL = CCL+CEL+CGL+CDL+CDDL+CLCL+CLHL+CSL+CRL+CPPL+RECLAM+WELLL+
     1     RCL+DICHAR
      CM = CGM+CLHM+CRM+CDDM+CSM+CPPM+WELLM+RCM
      CML = CL + CM
      IF(ITAXFR.EQ.1) WRITE(LD,2950) CL,CM,CML
      CTAX = CM * TXFRRT/100
      CLT = CL
      CMT = CM
 2950 FORMAT(T8,'SUBTOTAL DIRECT',T40,-3PF8.0,T50,F8.0,T60,F8.0)
      IF(ITAXFR.EQ.1) WRITE(LD,3000)  CTAX,CTAX
 3000 FORMAT(T8,'TAX AND FREIGHT',T50,-3PF8.0,T60,F8.0/T8,60('-'))
      IF(ITAXFR.EQ.1) CMT = CMT + CTAX
      CMTL = CLT + CMT
      WRITE(LD,3050) CLT,CMT,CMTL
```

```
 3050 FORMAT(T8,'TOTAL DIRECT LANDFILL INVESTMENT',T40,-3PF8.0,
     1       T50,F8.0,T60,F8.0/ T8,60('-'))
C
C      REPLACE TVA INDIRECT CALCULATIONS WITH IAPCS CALCULATIONS
C
C       CP = CMTL
C       CPTOT=CP
C      IF(ITAXFR.EQ.1) CPTOT=CP+CTAX
C      WRITE(LD,3100)
C3100 FORMAT(T8,'INDIRECT INVESTMENT',/T8,60('-')//)
C       IF (IECON.EQ.1) GO TO 1011
C      OHR=(1.0+(PENGIN+PARCH+PFLDEX+PFEES+PCONT)/100.0)
C     &     *(1.0+(PSTART+PCONIN)/100.0)
C       GO TO 1021
C 1011 CONTINUE
C       OHR= (1.0+(PENGIN+PARCH+PFLDEX+PFEES)/100.0)
C      &     *(1.0+PCONT/100.0)
C     &     *(1.0+(PSTART+PCONIN)/100.0)
C 1021 CONTINUE
C       CPO = CDET+ OHR * CMTL
C       PE = CMTL * PENGIN/100/1000
C       PA = CMTL * PARCH/100/1000
C       PF = CMTL * PFLDEX/100/1000
C       PFEE = CMTL * PFEES/100/1000
C       PI = PE + PA + PF + PFEE
C       PCN = (PI+CDET/1000 + CMTL/1000) * PCONT/100
C       IF(IECON.NE.1) PCN = (PCONT/100)*((CMTL/1000)+(CDET/1000))
C       TFCL =(CMTL/1000) + PI + PCN + CDET/1000
C       IF(IRPT.NE.0) WRITE(LD,3150) PENGIN,PE,PARCH,PA,PFLDEX,PF,PFEES,
C      1PFEE,PCONT,PCN,TFCL
C 3150 FORMAT(T8,'ENGINEERING DESIGN AND SUPERVISION (',F4.1,' )',T60,
C      1  F8.0/T8,'ARCHITECT AND ENGINEERING CONTRACTOR(',F4.1,' )',T60,
C      2  F8.0/T8,'CONSTRUCTION EXPENSES (',F4.1,' )',T60,F8.0/
C      3        T8,'CONTRACTOR FEES (',F4.1,' )',T60,F8.0/
C      4        T8,'CONTINGENCY (',F4.1,' )',T60,F8.0/
C      5        T8,60('-')/T8,'TOTAL FIXED INVESTMENT',T60,F8.0/)
C       WRITE(LD,3200) CA
C 3200 FORMAT(T8,'LAND COST',T60,-3PF8.0/)
C       IF(IRPT.EQ.0) WRITE(LD,3225) TFCL
C 3225 FORMAT(T8,'LANDFILL SITE',T60,F8.0//)
       YD3 = TYD3
       WRITE(8,*) ' FILAND ='
       FILAND = AREA(9)
       ELECLF =((SITE / 400000 ) * 6000 ) + EL + ELP
       WATERF =(OPP/5500) * (15.0 * YD3) / 0.14
       FILANL =(OPP/5500) * 19.5 * XLT / 21
       PNDMAT(7) = CM
       PNDLAB(7) = CL
       POND(7) = CML
       ELAN = CDM
       WRITE(LD,4000)
 4000 FORMAT(T8,'REVENUE QUANTITIES',/T8,'------- ----------'//)
```

```
      WRITE(LD,4100) TWLF,DISFUL,ELECLF,WATERF,FILANL
 4100 FORMAT(T8,'LANDFILL LABOR',T26,F6.0,T34,'MAN-HRS',/
     1       T8,'DIESEL FUEL',T25,F7.0,T34,'GALLONS',/
     2       T8,'ELECTRICITY',T24,F8.0,T34,'KWH',/
     3       T8,'WATER',T26,F6.0,T34,'K-GALLONS'/
     4       T8,'ANALYSIS',T27,F5.0,T34,'MAN-HRS'///)
      CAPCST(10,1)=CMTL+CDET
C>>>  SUBTRACT ON-SITE WASTE FROM DISPOSAL QUANTITIES
  888 CONTINUE
      ANN(12)= ANN(12)*(1.-PNDCAP)
      ANN(13)= ANN(13)*(1.-PNDCAP)
C
      ANN(3)=ANN(3)+TWLF
      ANN(4)=ANN(4)+ELECLF
      ANN(5)=ANN(5)+WATERF
      ANN(20)=ANN(20)+DISFUL
      ANN(21)= ANN(21)+FILANL
C
      WRITE(8,*) ' ANN(10 '
      ANN(10)=ANN(10) + CAPCST(10,1)*P(238)*.01
C
      IF (TVA) THEN
       P(231) = INT((P(231)-1.) + 2.*(BSIZE-100.)/900. + .5)
       P(232) = INT((P(232)-1.) + 2.*(BSIZE-100.)/900. + .5)
      ENDIF
C
C     CALCULATE ENGINEERING DESIGN AND SUPERVISION (TVA) OR
C     GENERAL FACILITIES (EPRI)
C
      IC(1) = P(230)/100. * CMTL
      INDCST(1) = INDCST(1) + IC(1)
C
C     CALCULATE A-E CONTRACTOR(TVA) OR ENG. AND HOME OFF. FEES (EPRI)
C
      IC(2) = P(229)/100. * CMTL
      INDCST(2) = INDCST(2) + IC(2)
C
C     CALCULATE CONSTRUCTION EXPENSE (TVA) PROJECT CONTINGENCY (EPRI)
C
      IC(3) = P(231)/100. * CMTL
      INDCST(3) = INDCST(3) + IC(3)
C
C     CALCULATE CONTRACTOR FEES (TVA) OR PROCESS CONTINGENCY (EPRI)
C
      IC(4) = P(232)/100. * CMTL
      INDCST(4) = INDCST(4) + IC(4)
C
C     SUM DIRECT AND INDIRECT CAPITAL COSTS TO HAVE A BASE FOR THE
C     REMAINDER OF THE INDIRECT COSTS.
C
      COST = CMTL + IC(1) + IC(2) + IC(3) + IC(4)
C
```

```
C       CALCULATE CONTINGENCY (TVA) OR SALES TAX (EPRI)
C
        WRITE(8,*)  ' IF (TVA '
        IF (TVA) THEN
              IC(5) = P(233)/100. * COST
           ELSE
              IC(5) = P(233)/100. * CAPCST(10,1)
              COST= COST+IC(5)
           ENDIF
        INDCST(5) = INDCST(5) + IC(5)
C
C       INCLUDE EQUIPMENT FOR REMAINDER OF INDIRECTS
C
        COST= COST + CDET
C
C       INCLUDE IC(5) IN RUNNING COST BASE TOTAL
C
        COST = COST + IC(5)
C
C       CALCULATE ROYALTIES (TVA) OR ROYALTY ALLOWANCE (EPRI)
C
        IC(6) = P(234)/100. * COST
        INDCST(6) = INDCST(6) + IC(6)
C
C       CALCULATE INTEREST DURING CONSTRUCTION (TVA)
C
        IF (TVA) THEN
          IC(8) = P(235)/100. * COST
          INDCST(8) = INDCST(8) + IC(8)
        ENDIF
C
C         CALCULATE ALLOWANCE FOR STARTUP AND MODIFICATIONS (TVA) OR
C                   INITIAL CATALYST (EPRI)
C
        IC(9) = P(236)/100. * COST
        INDCST(9) = INDCST(9) + IC(9)
C
C         CALCULATE LAND (TVA AND EPRI)
C
        IC(10) = CA
        INDCST(10) = INDCST(10) + IC(10)
C
        ITRUCK = ITRUCK + 1
        TRULOD = TYD(II)
        RETURN
        END
```

```
$LARGE Q,LAB
C
      SUBROUTINE PARMED(ICH,LAB1)
C
C           THIS SUBROUTINE ALLOWS THE USER TO RESELECT, EDIT, PRINT-OUT
C      OR RECEIVE HELP ON THE IAPCS PARAMETER FILE.
C
C
C      WRITTEN BY    JEFF WINGET
C                    PEI ASSOCIATES, INC.
C                    (3/14/85) (MOD JP 8/25/85)
C
C
      INTEGER IYOC
      COMMON /PARM/P(500)
      DIMENSION ISKIP(10),CIPU(4,11),RAT(4),IRATE(50)
      DIMENSION Z(20),Z1(20),LAB(20,100),Q(20,100),LL(100),ICH(500),
     & LAB1(500),MAP(20,100)
      CHARACTER*64 LAB,BLANK,LAB1
      CHARACTER*4 EXT(2),ECF(2)
      CHARACTER*8 FILE2,FILEN,FILENCH,MONTH(12)*9
      CHARACTER*1 LS(8),B1,Q2,QUEST
      CHARACTER*12 FILE1,FILECH,MODNAM(14)
      INTEGER Y,Z,Z1,FORM,IAPCS(9),FMB,CT,BOTTOM,TYPCT,CLNLVL,YY(500)
      LOGICAL EX,TVA,KO(2),CHANGE(500),ANY,EC
      COMMON /PEDIT/ CHANGE,FILE1
      COMMON /INPUT/    BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,
     &          PNAROM,ALKROM,CLROM,FEOROM,PMROM,VROM,FCROM,HHVCLN,PSCLN,
     &                PACLM,CSTCLN,PNACLN,ALKCLN,CLCLN,FEOCLN,PMCLN,
     &          VCLN,FCCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,IAPCS,KO,
     &                TVA
      DATA BLANK/'
     &          '/,B1/' '/
      DATA EXT/'.TVA','.EPR'/,ECF/'TVA ','EPRI'/
      DATA FILEN/'PARMFILE'/
      DATA MODNAM/    'System Wide','Unctrl Coal ','Fan          ',
     &'Economic    ','LSD         ','Low Nox/Over','Fabr. Filter',
     &'ESP         ','LIMB        ','Spray Humid.','DSI          ',
     &'FGD System  ','FGD Econs   ','Cost based  '/
      DATA MONTH/'  JANUARY',' FEBRUARY','    MARCH','    APRIL',
     &            '      MAY','     JUNE','     JULY','   AUGUST',
     &           'SEPTEMBER','  OCTOBER',' NOVEMBER',' DECEMBER'/
C
      I=0
 1000 FORMAT (I3)
 1002 FORMAT(3X,F10.0,A64)
 1003 FORMAT (3x,F10.0,I1,A63)
 1001 FORMAT (I3,G10.4,A64)
 1010 FORMAT (' Edit parameters? (Y/N)>>')
 1020 FORMAT ('  0 - Exit parameter editor',/,'  1 - Change system',
     &' wide parameters',/,'  2 - Change uncontrolled coal parameters',
     &/,'  3 - Change fan parameters',/,'  4 - Change economic',
```

```
     &' parameters',/,'  5 - Change LSD parameters',/,'  6 - Change ',
     &'low nox burner/overfire air parameters',/,'  7 - Change ',
     &'fabric filter parameters',/,'  8 - Change ESP parameters',/,
     &'  9 - Change LIMB parameters',/,' 10 - Change spray ',
     &'humidification parameters',/,' 11 - Change DSI parameters',
     &/,' 12 - Change FGD parameters (refer to TVA Manual '
     &,'EPA-600/8-85-006, March,'
     &,' 1985)',/,' 13 - Change FGD economic ',
     &'parameters (IBID)',/,' 14 - Change cost based parameters',
     &//,'  Which option >>')
1025 FORMAT ('    0 Exit to editor menu')
1030 FORMAT (' ',I3,' ',A64,F10.4)
1031 FORMAT (' ',I3,' ',A64,F10.2)
1040 FORMAT ('  Which do you want to change >>')
1050 FORMAT (' Change to >>')
1060 FORMAT (' ',I3,' Go to next menu')
1070 FORMAT (' ',I3,' Go to previous menu')
1080 FORMAT (' Input parameter file name (',A4,') >>')
1090 FORMAT (' Write parameters to file (Y/N)>>')
1100 FORMAT (' 1) TVA or 2) EPRI format on output >>')
1110 FORMAT (' File ',A,' does not exist.'/' Type Y to create >>')
1120 FORMAT ('  1 - Print system',
     &' wide parameters',/,'  2 - Print uncontrolled coal parameters',
     &/,'  3 - Print fan parameters',/,'  4 - Print economic',
     &' parameters',/,'  5 - Print LSD parameters',/,'  6 - Print ',
     &'low nox burner/overfire air parameters',/,'  7 - Print ',
     &'fabric filter parameters',/,'  8 - Print ESP parameters',/,
     &'  9 - Print LIMB parameters',/,' 10 - Print spray ',
     &'humidification parameters',/,' 11 - Print DSI parameters',
     &/,' 12 - Print FGD parameters (refer to TVA manual,',
     & 'EPA-600/8-85-006, March, '
     &,' 1985)',/,' 13 - Print FGD economic parameters (IBID)',/,
     &' 14 - Print cost based parameters',//,'  Which option >>')
   2 FORMAT(A1)
     DO 7 I= 1,500
   7 CHANGE(I)= .FALSE.
   8 WRITE (6,1100)
     READ (5,*,ERR=8) FORM
     TVA=.TRUE.
     IF (FORM.EQ.2) TVA=.FALSE.
     IF (FORM .NE. 1 .AND. FORM .NE. 2) GO TO 8
  10 WRITE (FILE1,'(A8,A4)') FILEN,EXT(FORM)
     FILE2= FILEN
  90 OPEN (1,FILE=FILE1,STATUS='OLD')
     DO 11 I=1,20
  11  Z(I)= 0
     REWIND (1)
     READ (1,1) NP
   1 FORMAT (I4)
     ICONT=1
     DO 100 I=1,NP
        READ(1,1000) Y
```

```
          BACKSPACE 1
          IF (Y.NE.14) READ (1,1002) Q(Y,Z(Y)+1),LAB(Y,Z(Y)+1)
          IF (Y.EQ.14) READ (1,1003) Q(Y,Z(Y)+1),IRATE(ICONT),
     &    LAB(Y,Z(Y)+1)
          IF (Y.EQ.14) ICONT=ICONT+1
          MAP(Y,Z(Y)+1)=I
          YY(I)= Y
          LAB1(I)=LAB(Y,Z(Y)+1)
          P(I)= Q(Y,Z(Y)+1)
          Z(Y)=Z(Y)+1
  100 CONTINUE
C
C     PARAMETER FILE MENU
C
   20 CALL CLS
      WRITE(6,2000) ECF(FORM),FILEN
 2000 FORMAT(//T11,'Economic Format: ',A,/T11,' Parameter File: ',A)
      WRITE(6,2010)
 2010 FORMAT(/'  You may:',
     &       /T11,'1. Switch to another existing parameter file.'
     &       /T11,'2. Edit parameter file/create a new parameter file.'
     &       /T11,'3. Display parameter file explanation.'
     &       /T11,'4. Print out parameter group.'
     &       /T11,'5. Leave this menu and begin input sequence.'
     &       /T11,'6. Stop the program without making a run.'
     &      //T11,'Input the number of your selection>>')
      READ(5,*,ERR=20) IOPT
      IF(IOPT .LT. 1 .OR. IOPT .GT. 6) GO TO 20
      GO TO (24,101,28,32,10000,20000), IOPT
C>>>> SWITCH TO ANOTHER PARAMETER FILE
C>>>  ESTABLISH NEW FILE NAME
   24 WRITE(6,2020)
 2020 FORMAT('  Enter parameter file name>>')
      DO 25 I=1,8
   25 LS(I)= B1
      READ(5,'(8A1)',ERR=24) (LS(I),I=1,8)
      CALL FTRIM(LS,FILECH,EXT(FORM),FILENCH,EC)
      IF(EC) GO TO 24
C>>>  VERIFY EXISTENCE OF PARAMETER FILE
      INQUIRE(FILE=FILECH,EXIST=EX)
      IF(.NOT. EX) THEN
         WRITE(6,2030)
 2030    FORMAT('  This file does not exist.',/' If you want to create',
     &          ' a new parameter file, use option 2.')
         PAUSE
         GO TO 20
      ELSE
C>>      IF FILE EXISTS, ASSIGN TO PERMANENT FILE NAMES
         FILE1= FILECH
         FILEN= FILENCH
C>>>     GO BACK TO READ IN NEW FILE
         GO TO 90
```

```
      ENDIF
C>>>> PRINT HELP SCREEN FOR PARAMETER FILE
   28 CALL HELP('PARMHELP.DAT')
      GO TO 20
C>>>> PRINT PARAMETER FILE
   32 CALL CLS
      WRITE (6,1120)
      READ (5,*,ERR=32) IPRT
      IF(IPRT.LE.0) GOTO 20
      IF (IPRT.GT.14) GOTO 32
      IF(IPRT.EQ.14) CALL RATMOD(IYOC,IMOC,RAT)
      MS=IPRT
      WRITE(7,2040) FILE1
 2040 FORMAT(1H+,T36,A,/T35,14(1H=))
      WRITE (7,2045) MODNAM(IPRT)
 2045 FORMAT(/,T36,A\)
      IF (IPRT.EQ.14) WRITE(7,*) MONTH(IMOC),',  ',IYOC
      IF (IPRT.NE.14) WRITE(7,*)
      WRITE(7,2050)
 2050 FORMAT(/'     VALUE      DESCRIPTION'/' ---------      ',65(1H-))
      DO 30 I=1,Z(IPRT)
        IF(MOD(I,57) .EQ. 0) THEN
          WRITE(7,2060)
 2060     FORMAT(1H1,T69,'(CONTINUED)')
          WRITE(7,2040) FILE1
          WRITE(7,2050)
        ENDIF
        IF (LAB(IPRT,I).EQ.BLANK) GOTO 30
        IF (IPRT.NE.14) RAT(IRATE(I))=1.
        WRITE(7,2070) Q(IPRT,I)*RAT(IRATE(I)),LAB(IPRT,I)
 2070   FORMAT(1H ,G10.4,3X,A)
   30 CONTINUE
      WRITE(7,10010)
10010 FORMAT('1')
      GO TO 20
  101 DO 110 I=1,20
         DO 115 J=Z(I),1,-1
            IF (LAB(I,J).NE.BLANK) THEN
               Z1(I)=J
               GO TO 110
            ENDIF
  115    CONTINUE
  110 CONTINUE
      IWRITE= 0
      ION= 0
      MAX=14
  120 CALL CLS
      WRITE (6,1020)
C     READ IN MODULE NUMBER
      READ (5,*,ERR=120) MS
      IF (MS.LT.1) GOTO 400
      IF (MS.GT.MAX) GOTO 120
```

```
      IF (MS.EQ.14) CALL RATMOD(IYOC,IMOC,RAT)
      L=0
      INCRE=0
      ISKIPT=0
      K=1
      DO 185 ICONTR=1,10
      ISKIP(ICONTR)=0
  185 CONTINUE
  190 ITEMP1=0
  200 ITEMP=Z1(MS)
  210 IF (Z1(MS)-ISKIPT+INCRE.GT.ITEMP1+16) ITEMP=ITEMP1+16
  220 CALL CLS
      IF (ITEMP1.EQ.0) WRITE (6,1025)
      IF (ITEMP1.GT.1) WRITE (6,1070) ITEMP1
      ICONT=1
      DO 130 I=ITEMP1+1-INCRE+ISKIPT,Z1(MS)
         IF (LAB(MS,I).EQ.BLANK) GOTO 125
         L=L+1
         IF(MS .NE. 14) THEN
           RAT(IRATE(I))=1.
           WRITE (6,1030) L,LAB(MS,I),Q(MS,I)*RAT(IRATE(I))
         ELSE
           WRITE(6,1031) L,LAB(MS,I),Q(MS,I)*RAT(IRATE(I))
         ENDIF
         LL(L)=I
         IF (ITEMP.EQ.Z1(MS)) GOTO 130
         IF (L.GE.ITEMP-1) GOTO 135
  125 IF (LAB(MS,I).EQ.BLANK) ISKIP(K)=ISKIP(K)+1
  130 CONTINUE
      GOTO 145
  135 IF (Z1(MS)-ISKIPT+INCRE-ISKIP(K).GT.ITEMP) WRITE (6,1060) ITEMP
      IF (Z1(MS)-ISKIPT+INCRE-ISKIP(K).GT.ITEMP) L=L+1
  145 WRITE (6,*)
 7000 WRITE (6,1040)
      READ (5,*,ERR=7000) J
      IF (J.LE.0) GOTO 140
      IF (J.GT.Z1(MS)-ISKIPT+INCRE-ISKIP(K)) GOTO 7000
      IF (J.LE.ITEMP1) THEN
         K=K-1
         ISKIPT=0
         DO 137 ICONTR=1,K-1
            ISKIPT=ISKIPT+ISKIP(ICONTR)
  137    CONTINUE
         ISKIP(K)=0
         ITEMP=ITEMP1
         ITEMP1=ITEMP1-16
         L=ITEMP1
         INCRE=INCRE-1
         IF (ITEMP1.EQ.0) GOTO 190
         GO TO 210
      ENDIF
      IF (J.GE.ITEMP.AND.ITEMP.NE.Z1(MS)) THEN
```

```
            DO 450 J=1,Z(I)
               WRITE (LAB1(MAP(I,J)),'(A1,A63)') CHAR(48+IRATE(J)),
     &         LAB(I,J)
  450       CONTINUE
  410    CONTINUE
         IF (IWRITE .NE. 1) GOTO 20
         ICONT=1
         DO 430 I=1,NP
  430       WRITE(1,1001) YY(I),P(I),LAB1(I)
         GOTO 20
10000    CONTINUE
         DO 10011 I=1,20
           DO 10011 J=1,Z(I)
             LAB1(MAP(I,J))= LAB(I,J)
10011    CONTINUE
         RETURN
20000    STOP
         END
C
         SUBROUTINE FTRIM(LS,FILECH,EXT,FILEN,EC)
C
C        THIS SUBROUTINE CONVERTS A SINGLE CHARACTER 8 ELEMENT ARRAY AND
C        A VALID FOUR CHARACTER EXTENSION INTO A VALID DOS FILE NAME BY
C        REMOVING EXTRANEOUS BLANKS. IT DOES NOT CHECK FOR INVALID
C        CHARACTERS. IF EC IS FALSE, AN ERROR WAS DETECTED IN THE
C        CONVERSION.
C
C        WRITTEN BY PEI ASSOCIATES (8-25-85 JP)
C
         CHARACTER*1 LS(8),B1,EXT*4,FILECH*12,FMT*12,FILEN*8
         LOGICAL EC
         EC= .FALSE.
         B1= ' '
C>>      DETERMINE LENGTH OF FILE NAME
         DO 26 I=1,8
         IF(LS(I) .EQ. B1) GO TO 27
   26    CONTINUE
         I= 9
   27    J= I-1
C>>      WRITE FILE NAME FORMAT TO BUFFER.(J IS FNAME LENGTH REPEAT COUNT)
         WRITE(FMT,'(1H(,I1,6HA1,A4))') J
C>>      SET DESIRED FILE NAME+EXT (FILECH)
         WRITE(FILECH,FMT,ERR=99) (LS(I),I=1,J),EXT
         WRITE(FILEN,'(8A1)') (LS(I),I=1,J)
         RETURN
   99    EC=.TRUE.
         RETURN
         END
```

```
      ISKIPT=0
      DO 127 ICONTR=1,K
         ISKIPT=ISKIPT+ISKIP(ICONTR)
  127 CONTINUE
         K=K+1
         ISKIP(K)=0
         ITEMP1=ITEMP
         INCRE=INCRE+1
         GO TO 200
      ENDIF
      M=LL(J)
 7001 WRITE (6,1050)
      READ (5,*,ERR=7001) Q(MS,M)
      IF (MS.EQ.14) Q(MS,M)=Q(MS,M)/RAT(IRATE(M))
      ISKIP(K)=0
      ION=1
      CHANGE(MAP(MS,M))= .TRUE.
      L=ITEMP1
      GOTO 210
  140 GOTO 120
  400 IF (ION.EQ.0) GOTO 401
      WRITE(6,1090)
      READ(5,2,ERR=400) QUEST
      IF(QUEST.NE.'Y'.AND.QUEST.NE.'y'.AND.QUEST.NE.'N'.AND.
     & QUEST.NE.'n') GO TO 400
       IF (QUEST .EQ. 'Y'.OR.QUEST.EQ.'y') THEN
         DO 403 I=1,8
  403      LS(I)= B1
         WRITE(6,2020)
         READ(5,'(8A1)',ERR=400) (LS(I),I=1,8)
         CALL FTRIM(LS,FILECH,EXT(FORM),FILENCH,EC)
         INQUIRE(FILE=FILECH,EXIST=EX)
         IF(EX) THEN
  405      WRITE(6,4090)
 4090      FORMAT('  This file already exists.'/' Do you want to'
     &           ' overwrite it? >>')
           READ(5,2,ERR=405) Q2
           IF(Q2.NE.'Y'.AND.Q2.NE.'N'.AND.Q2.NE.'y'.AND.Q2.NE.'n')
     &     GO TO 405
           IF(Q2 .EQ. 'N'.OR.Q2.EQ.'n') GO TO 400
         ENDIF
         FILE1= FILECH
         FILEN= FILENCH
         IWRITE= 1
       ENDIF
  401 IF (IWRITE.EQ.1) OPEN (1,FILE=FILE1,STATUS='NEW')
      IF (IWRITE.EQ.1) WRITE (1,1) NP
      DO 410 I=1,20
         DO 420 J=1,Z(I)
            P(MAP(I,J))=Q(I,J)
  420    CONTINUE
         IF (I.NE.14) GOTO 410
```

```
$DEBUG
C
C              INTEGRATED AIR POLLUTION CONTROL SYSTEM PROGRAM
C              ========== === ========= ======= ====== =======
C
C                LIMESTONE INJECTION MULTI-STAGE BURNER (LIMB) ROUTINE
C                --------- --------- ---------- ------ ------ -------
C
C
C      WRITTEN BY:     J. R. SCHOTTELKOTTE
C                      PEDCO COMPUTER SERVICES
C                      JUNE, 1983
C
C      WRITTEN FOR:    G. E. BAKER
C                           PEI ASSOC., INC.
C                           PEI PN 3568-3
C                      J. O. MILLIKEN
C                           U. S. EPA:   IERL-RTP
C
C
C            THIS ROUTINE CALCULATES DESIGN PARAMETERS, CAPITAL AND
C       ANNUAL OPERATING COSTS, AND MATERIAL BALANCE FOR L/S INJECTION
C       MULTI-STAGEBURNER (LIMB) SYSTEMS. THE ALGORITHMS WERE DEVELOPED
C       AND REVIEWED BY G. E. BAKER OF PEI ASSOC., INC.
C
C
      SUBROUTINE LIMB
      LOGICAL KO(2),TVA,IESP
      INTEGER BOTTOM,CLNLVL,CT,FMB,TYPCT
      REAL IC,INDCST,SORBT,MPF,NOXEFF,MOL,MUNUSD
      CHARACTER*20 TYPE(8)
      DIMENSION AL(6),AH(6),BL(6),BH(6),CL(6),CH(6)
      DIMENSION AC(25),CC(5),IC(10),MWGT(6)
      COMMON /COSTS/ CAPCST(10,8),ANN(25),INDCST(10)
      COMMON /COAL1/ HHV,PS,PA,CST,PNA,ALK,CL1,FEO,PM
      COMMON /STREAM/ PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
      COMMON /INPUT/BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,HHVCLN,PSCLN,PACLN,CSTCLN,ALKCLN,PNACLN,
     &CLCLN,FEOCLN,PMCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,IAPCS(9),KO
     &,TVA
      COMMON /INSYSP/ CCEFF,UTEFF,BNHR,FANLOS,QIN,ALKIN
      COMMON /PARM/ P(500)
      COMMON /MATBAL/ TABLE(8,8)
      COMMON /OPT/ BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
      COMMON /FAN/ DELTAP,CSTFAN
      COMMON/DUM/VERGPM(5),VERHD(5),VTOTLC(5),VTOTMC(5),VERTHP(8),
     1   AGITHP(10),AGITLC(10),AGITMC(10),DIATNK(10),HTTNK(10),
     2   TNKLC(10),TNKMC(10),VOLTNK(10),GPMPMP(11),PMPHP(11),
     3   TPMPLC(11),TPMPMC(11),TPH3,HPVAC,THKHT,CMECLC,CMECMC,CON3LG,
     4   CON8LG,DIASIL,DISPHD,FILTAR,HPMILL,HPSLAK,HTELE2,HTSILS,
     5   RAKEHP,REHTLC,REHTMC,SOBLLC,SOBLMC,SOBM,TFILTL,TFILTM,THIKAR,
     6   THKDIA,TPHSLK,TOTTKL,TOTTKM,VOLSIL,SOBL,NOTNK(10),NOMOT(45),
```

```
     7  NOPMOT(45),NSPMOT(45),NGR18,NOFILT,NOPREP,NOSILO,NOTHIK,NSFILT
      COMMON/DUMM/PMPFDN(11),TNKFDN(10),SPGR(10),BLDLAB(9),BLDMAT(9),
     1 EQULAB(9),EQUMAT(9),FDNLAB(9),FDNMAT(9),STRLAB(9),STRMAT(9),
     2 FBLDSF,FILFDL,FILFDM,HPFDRV,HPFPMP,VSCTNK,VERFDN,ELE2L,ELE2M,
     3 CSLAKL,CSLAKM,GPMILL,PCONL1,PCONL2,PCONL4,PCONM1,PCONM2,PCONM4,
     4 PEQLC1,PEQLC2,PEQMC1,PEQMC2,PSTLL1,PSTLL2,PSTLL3,PSTLL4,PSTLM1,
     5 PSTLM2,PSTLM3,PSTLM4,TPSLAK,TSLKHP,CMILLL,CMILLM,GPMTRN,NSUMPR
      COMMON/BIG1/BAGDLP,BAGRAT,BAGCOST,BGLIFE,PSSO2X,EBYPAS,RS,PSC,
     1 CCSA,EFFPS,ESPDLP,RESIST,SCARAT,FANHP,FANLC,FANMC,DELTA1,FILTSA,
     2 PSF,FILRAT,H,TL,VD,VE,TGASID,SKLAND,DRAINM,DRAINL,CLAYSK,WPSO3,
     3 WPSO4,WPCO3,WPINS,ASHCLN,WINDEX,PDMAX,FLOOD,SRAIR,OXAIR,OXSCFM,
     4 PRES,SO2RT,SO2RV,SO2RST,SDFEE,VPD,VT,HVCLN,HPTONW,
     5 ISLUDG,IFIXS,JSR,IVPD,IT,LON,LGMAX,ISTAND,NSBR,
     6 KEPASS,KPASO2,ICLAR,JSSO,JDIS,JSTR,JGPM,IRH,NSBF
      COMMON/CSTNDX/ICEPYR,CEPNDX,INVYR,IREVYR,RATLAB,RATMAT,ARM,ARL
      COMMON/EQCOST/NOP(100),PEL(100),PEM(100),TOTPEL(9),TOTPEM(9)
      COMMON/FLOWS/GPM,MFLOW,WFLOW,TOTFLO
      REAL MFLOW(20,12),WFLOW(20,12),TOTFLO(20),GPM(20)
      COMMON/IND/ENGIN,ARCTEC,FLDEXP,FEES,CONT,START,CONINT,FILMAT(9)
      COMMON/LS/IALK,ISR,SR,SRADD,WPMGO,WPI,WPM,LSADD,LIMEAD,MGOADD,WMG
      REAL LSADD,LIMEAD,MGOADD
      COMMON /ESC/ IYC,RATPLT,RATLA2,RATMA2,RATOM
      COMMON/TRAN/NOTRAN,NOREDN,NTOTAL,PCNTRN,NORAW,NOUTIL,NOUNIT,
     1            NOPROD,NOSHIP,NSPREP,NSCOMP,NOCOMP,NOBYDU
C
      DATA TYPE/'CALCITE.                ','DOLOMITE.               ',
     &'CALCITIC HYDRATE.    ','DOLOMITIC HYDRATE.   ',
     &'DPH.                 ','CPH.                 ',
     &'LIMESTONE.           ','LIME.                '/
      DATA MWGT/100,184,74,132,74,132/
C
C     CAPTURE EFFICIENCY COEFFICIENTS
C
      DATA CL/14.0750,16.0500,22.9250,23.1000,35.1500,32.3000/
      DATA CH/13.3500,12.7000,32.0000,32.0500,42.5000,40.4000/
      DATA BL/10.2250,17.7500,16.4750,29.2500,18.8750,34.5000/
      DATA BH/12.3250,22.7500,16.6000,30.2500,18.7750,32.5000/
      DATA AL/-0.0100,-0.0120,-0.0220,-0.0220,-0.0270,-0.0230/
      DATA AH/-0.0090,-0.0080,-0.0300,-0.0290,-0.0290,-0.0230/
C
C
C
      IO6 = 6
      IO7 = 7
      DO 10 I=1,25
      AC(I)= 0.0
   10 CONTINUE
      DO 15 I= 1,5
      CC(I)=0.0
   15 CONTINUE
      DO 20 I=1,10
      IC(I)=0.0
```

```
   20 CONTINUE
C*****************************************************************************
C     PART 1:  SIZING CALCULATIONS.
C
C     THIS PART CALCULATES SIZING PARAMETERS FOR USE LATER IN THE
C     PROGRAM.
C
C     THE FOLLOWING ARE POSSIBLE VALUES FOR ISTYPE:
C       1- CALCITIC LIMESTONE
C       2- DOLOMITIC LIMESTONE
C       3- CALCITIC HYDRATE
C       4- DOLOMITIC HYDRATE
C       5- CALCITIC PRESSURE HYDRATE
C       6- DOLOMITIC PRESSURE HYDRATE
C
C     SET DEFAULT VALUE FOR STOICHIOMETRIC RATIO (PARAMETER 111) AND
C     ISTYPE - CL,DL,CH,DH (PARAMETER 113)
C
      SR = P(111)
      ISTYPE = P(113)
C>>   AFTER PROCESSING, LIME(ISTYPE=8) AND LIMESTONE(ISTYPE=7) =1 & 3
      IF(ISTYPE .EQ. 7) ISTYPE= 1
      IF(ISTYPE .EQ. 8) ISTYPE= 3
      QR= P(125)
         IF (SR.GT.2 .AND. MOD(ISTYPE,2) .EQ. 0) THEN
            WRITE(6,*) 'WARNING: LIMB '
            WRITE (6,*) 'Ca/S RATIO EXCEEDS LIMIT. HAS BEEN RESET TO 2'
            SR=2
         ENDIF
         IF (SR.GT.4 .AND. MOD(ISTYPE,2) .EQ. 1) THEN
            WRITE(6,*) 'WARNING: LIMB '
            WRITE (6,*) 'Ca/S RATIO EXCEEDS LIMIT. HAS BEEN RESET TO 4'
            SR=4
         ENDIF
      EFLSO2= AL(ISTYPE)*QR+BL(ISTYPE)*SR+CL(ISTYPE)
      EFHSO2= AH(ISTYPE)*QR+BH(ISTYPE)*SR+CH(ISTYPE)
      WRITE(8,*) ' AL,BL,CL,ISTYPE',AL,BL,CL,ISTYPE
      EFFSO2= EFLSO2+P(123)*(EFHSO2-EFLSO2)
C
C      IF THIS IS AN OPTIMIZATION ASSIGN THE NEW EFFICIENCY AND CALCU-
C     LATE THE STOICHIOMETRIC RATIO. OTHERWISE SKIP TO LIMESTONE
C     USAGE.
C
      IF (EFFLIM .EQ. 0.) GO TO 100
      EFFSO2 = EFFLIM
      SRL= -(AL(ISTYPE)*QR+CL(ISTYPE)-EFFSO2)/BL(ISTYPE)
      SRH= -(AH(ISTYPE)*QR+CH(ISTYPE)-EFFSO2)/BH(ISTYPE)
      SR= SRL + (1.-P(123)) * (SRH-SRL)
C
C     DETERMINE LIMESTONE USEAGE RATE IN TONS PER HOUR. NOTE, PARA-
C     METER 114 IS THE LIMESTONE PURITY.
C
```

```
  100 TEMP1 = SR * (PPHSO2/2000.) * (1./(P(130 + ISTYPE)/100.))
      SORBT = MWGT(ISTYPE)/64. * TEMP1
      TOP = P(137)
      MOL = .96
      IF(ISTYPE.EQ.1.OR.ISTYPE.EQ.3.OR.ISTYPE.EQ.5) MOL = .56
C
C     CALCULATE UTILIZATION FACTOR.
C
      UTIL = EFFSO2/SR*.01
C
C     CALCULATE MASS OF PRODUCT FORMED PER POUND SO2 REMOVED. PARA-
C     METER 128 IS THE FRACTION OF SULFITE (SO3) AND 129 IS THE FRACTION
C     TION OF SULFATE (SO4).
      AMPRT = PPHSO2 * EFFSO2 * 2.6875*.01
      MUNUSD = SORBT * 2000 * (1. - UTIL)
C
C     WRITE DEBUG.
C
C
C*************************************************************************
C
C     PART 2:  MODULE SPECIFIC OUTPUT.
C
C     THIS PART WITES THE MODULE SPECIFIEC OUTPUT FOR THE LIMB
C     SYSTEM.
C
C     WRITE HEADER.
C
      IF(KO(1)) WRITE (IO6,6200)
      IF (KO(2)) WRITE(IO7,6200)
 6200 FORMAT
     &(///T5,'LIMESTONE INJECTION MULTI-STAGE BURNER (LIMB) SYSTEM')
C
C     REPRESENT EFFICIENCIES AS PERCENTS INSTEAD OF DECIMALS.
C     P(20) IS BOILER LOAD
C
      SO2EFF = EFFSO2
      X = P(20)
C
C     WRITE DETAIL MODULE SPECIFIC OUTPUT.
C
      IF (KO(1)) WRITE(IO6,6210) SR,TYPE(INT(P(113))),SO2EFF,X
      IF (KO(2)) WRITE (IO7,6210) SR,TYPE(INT(P(113))),SO2EFF,X
 6210 FORMAT( T10,'THE LIMB SYSTEM IS DESIGNED FOR A CA/S RATIO OF',
     &              F5.2,' AND MAKES USE OF'/T10,'THE SORBENT ',A20,
     &              'THE SO2 REMOVAL EFFICIENCY IS ',F5.1,'%.',
     &        /T10,'THE LIMB SYSTEM REPRESENTS OPTIMAL SORBENT',
     &        1X,'INJECTION AND ASSUMES ',F5.1,'%'/T10,'BOILER LOAD.',
     &        /T10,'THE COSTS ASSOCIATED WITH THE LIMB SYSTEM ',
     &        'INCLUDE SORBENT HANDLING,'/T10,'PREPARATION, AND ',
     &        'INJECTION; GAS HANDLING; WASTE HANDLING AND ',
     &        'DISPOSAL.')
```

```
      IF(NR .EQ. 2 .AND. KO(1)) WRITE(IO6,6212)
      IF(NR .EQ. 2 .AND. KO(2)) WRITE(IO7,6212)
 6212 FORMAT(T10,'ECONOMIZER MODIFICATIONS; SOOTBLOWER UPGRADES',
     &        '  AND SO3 CONDITIONING',/T10,'ARE ALSO',
     &        ' INCLUDED.')
      IF(EFFLIM .EQ. 0) THEN
         IF(KO(1)) WRITE(IO6,6215) P(125),EFLSO2,EFHSO2
         IF(KO(2)) WRITE(IO7,6215) P(125),EFLSO2,EFHSO2
 6215 FORMAT(T10,'THE RANGE OF VALID REMOVAL EFFICIENCIES FOR THIS'
     &          ,' CA/S RATIO AND A QUENCH'/T10'RATE OF',F5.0,'F/SEC'
     &          ,' IS ',F5.1,' TO ',F5.1,' %.')
      ENDIF
C
C*******************************************************************************
CC
C     PART 3:  DIRECT CAPITAL COST CALCULATIONS.
C
C    THIS PART CALCULATES THE DIRECT CAPITAL COST FOR THE LIMB
C     SYSTEM. RETROFIT IS AN IMPORTANT FACTOR IN THESE COST CALCU-
C     LATIONS.  IF NR=1, NEW CONSTRUCTION; IF NR=2, RETROFIT.
C
C     CALCULATE THE COST FROM LSPREP (CL OR DL) OR FROM LIMEPR
C     (CH OR DH)
C
      COSTM= 0.
      COSTL= 0.
      IF(P(113).EQ.7.) THEN
        TMPRATM = RATMAT
        TMPRATL = RATLAB
        RATMAT = 1.
        RATLAB = 1.
        NSPREP = 1
        LSADD = 2000 * SORBT
        WINDEX = 10.
        HPTONW = 5.7
        NSUMPR = 1
        CALL LSPREP(0.,0.,0.,0.)
        RATMAT = TMPRATM
        RATLAB = TMPRATL
        DO 200 III=1,28
         COSTL = COSTL + PEL(III)
         COSTM = COSTM + PEM(III)
  200   CONTINUE
      ELSEIF(P(113).EQ.8.) THEN
        TMPRATM = RATMAT
        TMPRATL = RATLAB
        RATMAT = 1.
        RATLAB = 1.
        NSPREP = 1
        LIMEAD = 2000 * SORBT
        CALL LIMEPR
        RATMAT = TMPRATM
```

```
        RATLAB = TMPRATL
        DO 300 III=1,23
         COSTL = COSTL + PEL(III)
         COSTM = COSTM + PEM(III)
  300   CONTINUE
      ELSE
        IF(SORBT .LE. 0.) THEN
          COSTM= 0.
        ELSEIF(SORBT*2000. .LT. 26100.) THEN
          COSTM= 62.85*SORBT*2000. + 871063. + 1366699.
        ELSEIF(SORBT*2000. .LE. 56749.) THEN
          COSTM= 62.85*SORBT*2000. + 871063. +1640039.
        ELSEIF(SORBT*2000. .LE. 170000.) THEN
          COSTM= 15.994*SORBT*2000. + 1184098. + 1776709.
        ELSE
          COSTM= 27.1*SORBT*2000. - 703968. + 2050049.
        ENDIF
        COSTM= COSTM*.5
      ENDIF
      CC(1) = COSTL + COSTM
      IF(NR .EQ. 1) THEN
C
C     CALCULATE THE COST OF THE ECONOMIZER (CC(2)). THIS FORMULA
C     VARIES BASED ON LIMESTONE RATE (SORBT) . CALCULATE THE BASE
C     COST FOR A RETROFIT AND ADJUST IF NEW.
        IF (SORBT .LE. 85.) CC(2) = (.02*SORBT + 1.89) * BSIZE *1000.
        IF (SORBT .GT. 85.) CC(2) = (SORBT/250.) * 10.95 *BSIZE *  1000.
        CC(2) = 0.25 * CC(2)
C
C      CALCULATE THE COST OF SOOT BLOWERS (CC(3))). AGAIN, THIS FORMU-
C     LA VARIES ON THE LIMESTONE RATE (SORBT). AGAIN, CALCULATE THE
C     BASE COST FOR A RETROFIT AND ADJUST IF NEW.
C
        IF (SORBT .LT. 45.) CC(3) = BSIZE * 1000.
        IF (SORBT .GE. 45.) CC(3) = (SORBT/250.) * 5.85 * BSIZE * 1000.
        CC(3) = .25 * CC(3)
C
C     CALCULATE COST OF UPGRADES
C
      ELSE
C>>>    ECONOMIZER MODIFICATION.
        IF(SORBT .LT. 85.) THEN
          ECFACT= 4.14*BSIZE*1000.
        ELSE
          ECFACT= (2.6696E-05*(SORBT*2000.-170000.)+4.14)*BSIZE*1000.
        ENDIF
C>>>    ESP UPGRADE
        IESP= .FALSE.
        DO 330 I=1,9
  330     IF(IAPCS(I) .EQ. 5) IESP= .TRUE.
        IF(P(130) .NE. 0. .AND. IESP) THEN
          PTLIMB= TOP*(AMPRT+MUNUSD*MOL)
```

```
            PRTUPG= PTLIMB/PPHPRT * 20.18 * BSIZE * 1000.
            CAPCST(5,1)= PRTUPG*RATPLT
          ENDIF
C>>>      GAS CONDITIONING
          SO3UPG= 5.65*BSIZE*1000.
C
          CC(5)= ECFACT+SO3UPG
        ENDIF
C>>>    SOOTBLOWER UPGRADES
        IF(SORBT .LT. 45.) THEN
          STFACT= 1.09*BSIZE*1000.
        ELSE
          STFACT= (1.2142E-05*(SORBT*2000.-90000.)+1.09)*BSIZE*1000.
        ENDIF
        CC(5)= CC(5)+ STFACT
C
C       SUM THE COSTS FOR THE LIMB SYSTEM.
C
        CAPCST(1,1) = (CC(1) + CC(2) + CC(3) + CC(4) + CC(5)) * RATPLT
C
C*******************************************************************************
C
C       PART 4:  INDIRECT CAPITAL COST CALCULATIONS.
C
C       THE INDIRECT CAPITAL COSTS ARE GENERALLY A PERCENTAGE OF THE
C       DIRECT CAPITAL COSTS OR DIRECT PLUS INDIRECT CAPITAL COSTS.
C        THE PERCENTAGES ARE RETAINED IN THE PARAMETER FILE (PARAMETERS
C       116 THRU 121). EACH INDIRECT COST FOR EACH MODULE IS SUMMED
C       INTO THE ARRAY 'INDCST'.
C
C       CALCULATE ENGINEERING DESIGN AND SUPERVISION (TVA) OR
C                 GENERAL FACILITIES (EPRI)
C
        IC(1) = P(117)/100. * CAPCST(1,1)
        INDCST(1) = INDCST(1) + IC(1)
C
C       CALCULATE A-E CONTRACTOR (TVA) OR ENG. AND HOME OFF. FEES (EPRI)
C
        IC(2) = P(116)/100. * CAPCST(1,1)
        INDCST(2) = INDCST(2) + IC(2)
C
C       CALCULATE CONSTRUCTION EXPENSE (TVA) OR PROJECT CONTINGENCY (EPRI)
C
        IC(3) = P(118)/100. * CAPCST(1,1)
        INDCST(3) = INDCST(3) + IC (3)
C
C        CALCULATE CONTRACTOR FEES (TVA) OR PROCESS CONTINGENCY (EPRI)
C
        IC(4) = P(119)/100. * CAPCST(1,1)
        INDCST(4) = INDCST(4) + IC(4)
C
C        SUM DIRECT AND INDIRECT CAPITAL COSTS TO HAVE A BASE FOR THE
```

```
C       REMAINDER OF THE INDIRECT COSTS.
C
        COST = CAPCST(1,1) + IC(1) + IC(2) + IC(3) + IC(4)
C
C       CALCULATE CONTINGENCY (TVA) OR SALES TAX (EPRI)
C
        IF (TVA) THEN
              IC(5) = P(120)/100. * COST
           ELSE
              IC(5) = P(120)/100. * CAPCST(1,1)
           ENDIF
        INDCST(5) = INDCST(5) + IC(5)
C
C       INCLUDE IC(5) IN RUNNING COST BASE TOTAL
C
        COST = COST + IC(5)
C
C       CALCULATE ROYALTIES (TVA) OR ROYALTY ALLOWANCE (EPRI)
C
        IC(6) = P(121)/100. * COST
        INDCST(6) = INDCST(6) + IC(6)
C
C       CALCULATE INTEREST DURING CONSTRUCTION (TVA)
C
        IF (TVA) THEN
          IC(8) = P(129)/100. * COST
          INDCST(8) = INDCST(8) + IC(8)
        ENDIF
C
C         CALCULATE ALLOWANCE FOR STARTUP AND MODIFICATIONS (TVA) OR
C                   INITIAL CATALYST (EPRI)
C
        IC(9) = P(127)/100. * COST
        INDCST(9) = INDCST(9) + IC(9)
C
C         CALCULATE LAND (TVA AND EPRI)
C
        IC(10) = P(181)
        INDCST(10) = INDCST(10) + IC(10)
C
C       WRITE DEBUG.
C
C!        WRITE(IODB,8400) IC,INDCST
C! 8400 FORMAT(T10,'IC=',6F12.2,
C!     &        /T10,'INDCST=',4F12.2,
C!     &        /T17,4F12.2)
C
C*******************************************************************************
C
C       PART 5:  ANNUAL COST CALCULATIONS.
C
C       ANNUAL COSTS ARE REPORTED FOR THE ENTIRE SYSTEM (ALL MODULES).
```

```
C     THIS PART CALCULATES THE INCREMENTAL COST FOR THIS MODULE.
C     MOST PARAMETERS ARE KEPT BY QUANTITY. ALL VALUES ARE ON A
C     PER YEAR BASIS.
C
      AC(1)= P(115)*8760.
      ANN(1) = ANN(1) + AC(1)
      AC(2) = P(124)/100. * AC(1)
      ANN(1) = ANN(1) + AC(2)
C
C     CALCULATE ELECTRICITY REQUIREMENT IN KILOWATTHOURS.
C
      AC(4) = 10.25 * 8760. * CF * SORBT
      ANN(4) = ANN(4) + AC(4)
C
C     NO WATER REQUIRED
C
C     NO REHEAT REQUIRED.
C
C     REAGENTS
C
      IF(ISTYPE .EQ. 1 .OR. ISTYPE .EQ. 3) THEN
        IX= 7 +ISTYPE/3
      ELSEIF(ISTYPE .EQ. 2) THEN
        IX= 14
      ELSE
        IX= ISTYPE+11
      ENDIF
      IF(P(113) .GE. 7) IX=15+P(113)
      AC(IX)= SORBT * 8760. * CF
      ANN(IX)= ANN(IX)+ AC(IX)
C
C     CALCULATE MAINTENANCE LABOR AND MATERIAL COST AS A PERCENTAGE
C     (PARAMETER 126) OF THE DIRECT CAPITAL COST.
C
      ANN(10) = ANN(10) + P(126)/100. * CAPCST(1,1)
C
C     NO FUEL COST PREMIUM
C
C     ONLY DRY WASTES ARE PRODUCED. CALCULATE THIS DISPOSAL QUANTI-
C     TY IN TONS. PARAMETER 131 IS THE BOTTOM ASH FACTOR.
C
      AC(13) = (1. - TOP) * (AMPRT + (MUNUSD * MOL)) * 8760. * CF/2000.
      ANN(13) = ANN(13) + AC(13)
C
C     WRITE DEBUG.
C
C!      WRITE(IODB,8500) AC,ANN
C! 8500 FORMAT (T10,'AC=',7F10.2,
C!     &        /T13,6F10.2,
C!     &        /T13,6F10.2,
C!     &        /T13,6F10.2,
C!     &        /T10,'ANN=',7F10.2,
```

```
C!     &        /T13,6F10.2,
C!     &        /T13,6F10.2,
C!     &        /T14,6F10.2)
C
C*******************************************************************************
C
C     PART 6:  MATERIAL BALANCE CALCULATIONS.
C
C     AT THE END OF EACH MODULE, THE STREAM PARAMETERS CONTAINED IN
C     THE COMMON 'STREAM' ARE ADJUSTED FOR THE AFFECTS OF THE MODULE.
C
      FGACFM = (2000 * SORBT/MWGT(ISTYPE) * 359.1 * (460 + AHET)/492.)/60.
      ACFM = ACFM + FGACFM
C
C     CALCULATE NEW FLUE GAS ALKALINITY. PARAMETER 130 IS THE FLY
C     ASH FACTOR AND PARAMETER 132 IS THE DEADBURN FACTOR.
C
      PPHALK = PPHALK + TOP * (MUNUSD * MOL)
C
C     CALCULATE NEW FLUE GAS SO2. NOTE, THE CURRENT VALUE IS RETAIN-
C     ED AS 'OPHSO2' FOR LATER USE.
C
      OPHSO2 = PPHSO2
      PPHSO2 = (1. - EFFSO2*.01) * OPHSO2
C
C
      IF(ISTYPE.GT.2) PPHH2O=PPHH2O+2000.*SORBT*359.1/MWGT(ISTYPE)*
     & ((460. + AHET)/492.)/60.
C
C     CALCULATE NEW FLUE GAS PARTICULATES. NOTE, THE CURRENT VALUE
C     IS RETAINED AS 'OPHPRT' FOR LATER USE.
C
      OPHPRT = PPHPRT
      PPHPRT =PPHPRT + TOP * (AMPRT + (MUNUSD * MOL))
C
C     CALCULATE NEW FLUE GAS FLOW IN LBS/HR.
C
      PPH = PPH - OPHPRT + PPHPRT
      IF(ISTYPE.LT.3) PPH = PPH + 2000 * SORBT * 44./MWGT(ISTYPE)
      IF(ISTYPE.GT.2) PPH = PPH + 2000 * SORBT * 18./MWGT(ISTYPE)
C
C     NO CHANGE TO TEMPERATURE
C
C     WRITE DEBUG
C
C!      WRITE (IODB,8600) PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2
C! 8600 FORMAT (T10,'PPH,ACFM,TEMP,PPHMST=',4F14.4,
C!     &        /T10,'PPHALK,PPHPRT,PPHSO2=',3F14.4)
C
C*******************************************************************************
C
C     PART 7: CONCLUSION.
C
C!  700 WRITE(IODB,8700)
C! 8700 FORMAT(T2,'LEAVING SUBROUTINE LIMB.')
C
      RETURN
      END
```

```
C
C
C      WASTE PASS
C        PROGAM TO PASS WASTE DISPOSAL VARIABLES
C
C        BY JEFF WINGET
C          3/22/85
C
       SUBROUTINE WPASS
       COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
      1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
      2 PSF, FILRAT, H, TL, VD, VE, TGAS1D, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
      3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
      4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
      5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
      6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
       COMMON/COAL/HVC, WPC, WPH, WPO, WPN, WPSUL, WPCL, WPASH, WPH2O, SULO, ASHO
       COMMON/CSTNDX/ICEPYR, CEPNDX, INVYR, IREVYR, RATLAB, RATMAT, ARM, ARL
       COMMON/FLYASH/IASH, ASHUPS, ASHSCR, ASHPRU, ASHPRS, ASHLBU, ASHLBS
       COMMON/HGAS2/MPCT, LBMH, LBH, SCFM, HHG, HWG, CFR, BCFR, RCFR
       COMMON/PCOST/POND(9), PDACRE, TOPHRS, PNDCAP, PNDLAB(9), PNDMAT(9), ELAN
       COMMON/PND1/PDEPTH, PMXEXC, DISTPD, ILINER, CLAYIN, CLAYUC, HYPLMC,
      1            HYPLLC, PSAMAX, ACREC, APRAIN, SAPILE
       COMMON/SCRUBR/LG, V, PPMOUT, SO2R, ELIMIT, PR, PD, WPS, OX, PH, NSTAGE,
      1              NGRID, HS, ACS, ACSTRN
       COMMON/SOLDIS/CASO3, CASO4, CACO3, INSOLS, WDIS, PSD, DENS, DEND
       COMMON/OPPSCH/IOPSCH, IYROP, IA(50)
       REAL MPCT(6), LBMH(6), LBH(6)
       COMMON/LANDIS/FILAND, TWLF, DISFUL, ELECLF, WATERF, FILANL, OVLF
       DIMENSION SW(4), AREA(10), B(5), ULFC(7)
       DIMENSION B1(5)
       REAL LG
       REAL INSOLS
       COMMON/COMPLE/ICOMPL, IRPT, IRUN, NEXT, IEQPR, IWTBAL, ITM, INPOPT, NOPART
       COMMON/PCOSTS/PCLEAR, PDEXEC, SODM, SODL, PERIML, PERIMM, ROADMM, ROADLL,
      1 RECLAM
       COMMON/PNDARW/AB, AID, AIE, AOD, AS, DL, P, VEC, VETT, WELLM, WELLL, WPSITE
       COMMON/PNDIND/INDPND, PENGIN, PARCH, PFLDEX, PFEES, PCONT, PSTART, PCONIN
       COMMON/RRIND/XINT, PCTMNT, PDMNTP, XINFLA, PCTOVR, PCTADM, UNDCAP,
      1         PCTINS, UC, IECON
       REAL *8 XINT, PCTMNT, PDMNTP, XINFLA, PCTOVR, PCTADM, UNDCAP, PCTINS, UC(1
      1        8)
       DIMENSION CYD(7), TYD(7), FUELDT(7), BYD(5)
       COMMON/TAXES/ITAXFR, TXFRRT, TXRAT, FRRAT, SERVRT, ROYALT, IOTIME, OTRATE
       COMMON/WGAS/WACFM, WSCFM, TWG, ASHOUT, WASHOT, WMPCT, WLBMH, WLBH
       REAL WMPCT(6), WLBMH(6), WLBH(6)
       COMMON/WGCOMP/YWCO2, YWHCL, YWSO2, YWO2, YWN2, YWH2O, WVAPOR, NTRAIN
       REAL NTRAIN
       COMMON/SCRUB/ISCRUB, IFOX, IADD, VLG, VTR, IFAN
       COMMON/SKGAS/SSCFM, SACFM, TSK, SDUST, SWDUST, SMPCT, SLBMH, SLBH, TGAS
       REAL SMPCT(6), SLBMH(6), SLBH(6)
       COMMON/TRAN/NOTRAN, NOREDN, NTOTAL, PCNTRN, NORAW, NOUTIL, NOUNIT,
```

```
     1              NOPROD, NOSHIP, NSPREP, NSCOMP, NOCOMP, NOBYDU
      COMMON/SRHT/TSTEAM, HVS, VRH, WSTEAM, ARH, DPRH, URH, Q, NT, NB, PL, OD1,
     1              ARINCL, ARCORT, NINCLT, NCORTT, NINCLB, NCORTB, TEMPS
      DIMENSION ARH(6), DPRH(6), URH(6), NT(6), NB(6), OD1(6), ARINCL(6),
     1           ARCORT(6), NINCLT(6), NCORTT(6), NINCLB(6), NCORTB(6)
C
      WRITE (14) ACREC, ASHO, ASHPRU, CLAYIN, CLAYUC, DISTPD, HYPLLC, HYPLMC
      WRITE (14) IA, ILINER, IRPT, ISLUDG, ITAXFR, OX, PARCH, PCLEAR, PCONIN
      WRITE (14) PCONT, PDEPTH, PDEXEC, PENGIN, PERIML, PERIMM, PFEES, PFLDEX
      WRITE (14) PMXEXC, PNDCAP, PSD, PSTART, RATLAB, RATMAT, RCFR, RECLAM
      WRITE (14) ROADLL, ROADMM, SODL, SODM, TOPHRS, TXFRRT, UC, WDIS, WELLL
      WRITE (14) WELLM, WPASH, IFIXS
C
      WRITE (14) TGAS, TWG, VPD, NTRAIN, DELTAP, NOTRAN, NTOTAL, ISCRUB, NOREDN
      WRITE (14) PD, DPRH(2)
C
      RETURN
      END
```

```
$DEBUG
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C      SUBPROGRAM WORKCP
C      REVISION DATE 82-11-04
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       SUBROUTINE WORKCP
      1       (REVHP, DIRECT, WORCAP)
C
       COMMON/BIG1/BAGDLP, BAGRAT, BAGCOST, BGLIFE, PSSO2X, EBYPAS, RS, PSC,
      1 CCSA, EFFPS, ESPDLP, RESIST, SCARAT, FANHP, FANLC, FANMC, DELTAP, FILTSA,
      2 PSF, FILRAT, H, TL, VD, VE, TGASID, SKLAND, DRAINM, DRAINL, CLAYSK, WPSO3,
      3 WPSO4, WPCO3, WPINS, ASHCLN, WINDEX, PDMAX, FLOOD, SRAIR, OXAIR, OXSCFM,
      4 PRES, SO2RT, SO2RV, SO2RST, SDFEE, VPD, VT, HVCLN, HPTONW,
      5 ISLUDG, IFIXS, JSR, IVPD, IT, LON, LGMAX, ISTAND, NSBR,
      6 KEPASS, KPASO2, ICLAR, JSSO, JDIS, JSTR, JGPM, IRH, NSBF
       COMMON/DUM/VERGPM(5), VERHD(5), VTOTLC(5), VTOTMC(5), VERTHP(8),
      &    AGITHP(10), AGITLC(10), AGITMC(10), DIATNK(10), HTTNK(10),
      &    TNKLC(10), TNKMC(10), VOLTNK(10), GPMPMP(11), PMPHP(11),
      &    TPMPLC(11), TPMPMC(11), TPH3, HPVAC, THKHT, CMECLC, CMECMC, CON3LG,
      &    CON8LG, DIASIL, DISPHD, FILTAR, HPMILL, HPSLAK, HTELE2, HTSILS,
      &    RAKEHP, REHTLC, REHTMC, SOBLLC, SOBLMC, SOBM, TFILTL, TFILTM, THIKAR,
      &    THKDIA, TPHSLK, TOTTKL, TOTTKM, VOLSIL, SOBL, NOTNK(10), NOMOT(45),
      &    NOPMOT(45), NSPMOT(45), NGR18, NOFILT, NOPREP, NOSILO, NOTHIK,
      &    NSFILT
C
       COMMON/BOILER/MW, BHR, EXSAIR, RMW, PREC, SPASH, WPRITE, KCLEAN, SMRW, SMCL
       COMMON/LANDIS/FILAND, TWLF, DISFUL, ELECLF, WATERF, FILANL, OVLF
       COMMON/LIQ2/IW4, AD, ADDC, ADLOSS, AMGLOS
       COMMON/LS/IALK, ISR, SR, SRADD, WPMGO, WPI, WPM, LSADD, LIMEAD, MGOADD, WMG
       COMMON/FLOWS/GPM, MFLOW, WFLOW, TOTFLO
       COMMON/SRHT/TSTEAM, HVS, VRH, WSTEAM, ARH, DPRH, URH, Q, NT, NB, PL, OD,
      &ARINCL, ARCORT, NINCLT, NCORTT, NINCLB, NCORTB, TEMPS
       COMMON/SCRUB/ISCRUB, IFOX, IADD, VLG, VTR, IFAN
       COMMON/FIXCOS/FIXADD, DS, TFIXEQ, TFIXLB, HPFIX(8)
       COMMON/OPPSCH/IOPSCH, IYROP, IA(50)
       COMMON/TRAN/NOTRAN, NOREDN, NTOTAL, PCNTRN, NORAW, NOUTIL, NOUNIT,
      1            NOPROD, NOSHIP, NSPREP, NSCOMP, NOCOMP, NOBYDU
       COMMON/RRIND/XINT, PCTMNT, PDMNTP, XINFLA, PCTOVR, PCTADM, UNDCAP,
      1              PCTINS, UC, IECON
       COMMON/WRIT9/DSBCAP, DALAND, DWRCAP, DTOFLO, MGWATT, KBTU, DLBH3,
      & DWLBH3, DWPS, AQ, DDIREC, DPOND, IHVC
       REAL INDCST
       INTEGER FMB, CT, TYPCT, CLNLVL, BOTTOM
       LOGICAL KO(2), TVA
       COMMON/INPUT/BSIZE, CF, AHFX, AHXT, HHVROM, PSROM, PAROM, CSTROM, PNAROM,
      &ALKROM, CLROM, FEOROM, PMROM, VROM, FCROM, HHVCLN, PSCLN, PACLN, CSTCLN,
      &PNACLN, ALKCLN, CLCLN, FEOCLN, PMCLN, VCLN, FCCLN, FMB, NR, IS, CT, TYPCT,
      &CLNLVL, BOTTOM, NM, IAPCS(9), KO, TVA
```

```
      COMMON /COSTS/ CAPCST(10,8),ANN(25),INDCST(10)
      COMMON/PCOST/POND(9),PDACRE,TOPHRS,PNDCAP,PNDLAB(9),PNDMAT(9),ELAN
      REAL MW,LSADD,LIMEAD,MGOADD
      REAL MFLOW(20,12),WFLOW(20,12),TOTFLO(20),GPM(20)
      REAL *8 XINT,PCTMNT,PDMNTP,XINFLA,PCTOVR,PCTADM,UNDCAP,PCTINS,UC(1
     1       8)
      REAL*8 DSBCAP,DALAND,DWRCAP,DTOFLO,MGWATT,KBTU,DLBH3,
     &DWLBH3,DWPS,AQ(40),DDIREC,DPOND
      DIMENSION ARH(6),DPRH(6),URH(6),NT(6),NB(6),OD(6),ARINCL(6),
     &ARCORT(6),NINCLT(6),NCORTT(6),NINCLB(6),NCORTB(6)
C
      DIMENSION DIRECT(9),WORCAP(9)
C
C     WORKING CAPITAL COSTS (BASED ON 7000 OPERATING HOURS)
C     NEW PREMISES BASED ON FIRST YEAR OPERATING HOURS
C     ** CALCULATE OPERATING HOURS USING IAPCS CAPACITY FACTOR **
      OPRHRS = 8760.*CF
      IVEN=0
      MYEAR=0
      NMANHR =2
      IF (ISCRUB.GE.3) IVEN=1
C      WCFA=.0577
C      WCFB=.1346
C      IF(IECON.NE.1) GO TO 1000
C      WCFA = .0833
C      WCFB =.1250
C      WCFC = .03
C1000 CONTINUE
      IF(IALK.EQ.1) LIMEAD=0.0
      IF(IALK.EQ.2) LSADD=0.00
      TMWORK = 1.12519 * MW**0.451502
      IF(NOTHIK.GT.0) MYEAR = 3
      IF(NOFILT.GT.0) MYEAR = 8
      TMWORK = TMWORK + MYEAR + IVEN
      AQ(1)=LSADD*OPRHRS/(2000.*1000.)
      AQ(2)=LIMEAD*OPRHRS/(2000.*1000.)
      IF(IFIXS.GT.0) AQ(2) = AQ(2) + FIXADD*OPRHRS/(2000.*1000.)
      AQ(3)=AMGLOS*OPRHRS/2000.
      AQ(4)=ADLOSS*OPRHRS/2000.
      AQ(NORAW+1) = TMWORK*2080. * (OPRHRS/8760) ** 0.5
      IF(IFIXS.GT.0) AQ(NORAW+1) = AQ(NORAW+1) + 10950.
      AQ(NORAW+2) = TWLF
      AQ(NORAW+3)=WSTEAM*OPRHRS/1000.
      AQ(NORAW+4)=(GPM(5)*60.*OPRHRS/1000.)+WATERF
      AQ(NORAW+5)=(((REVHP*.7457)+((100.+10*IVEN+5*NSPREP+(5+3*IVEN)*
     1       NOREDN)*(MW/500.)**.6))*OPRHRS)+ELECLF
      AQ(NORAW+6)=DISFUL
      IF(MW.GE.500) NMANHR = 3
      IF(MW.GT.750) NMANHR = 4
      AQ(NORAW+7) =((NMANHR * 2080.) *(OPRHRS/8760) ** 0.5)+ FILANL
      IF(IFIXS.GT.0) AQ(NORAW+7) = AQ(NORAW+7) + 750.
C
```

```
C      IAPCS INTERFACE TO FGD ANNUAL COSTS
C
C      NOTE: WASTE QUANTITIES (DIESEL FUEL, LABOR, AND ANALYSIS) ARE
C            INCLUDED IN THE WASTE MODULE.
C
       ANN(7)= ANN(7)+AQ(1)*1000.
       ANN(8)= ANN(8)+AQ(2)*1000.
       ANN(18)= ANN(18)+AQ(3)
       ANN(19)= ANN(19)+AQ(4)
       ANN(1)= ANN(1)+AQ(NORAW+1)
       ANN(6)= ANN(6)+AQ(NORAW+3)
       ANN(5)= ANN(5)+AQ(NORAW+4)
       ANN(4)= ANN(4)+AQ(NORAW+5)
       ANN(21)= ANN(21)+AQ(NORAW+7)-FILANL
C       WCMNT=(PCTMNT*(DIRECT(8)-DPOND/1000)+(PDMNTP*DPOND/1000))*10.
C       WCLS=AQ(1)*UC(1)*1000.
C       WCLIM=AQ(2)*UC(2)*1000.
C       WCMGO=UC(3)*AQ(3)
C       WCACID=UC(4)*AQ(4)
C       WCLAB=AQ(NORAW+1)*UC(NORAW+1)
C       WCLF =AQ(NORAW+2)*UC(NORAW+2)
C       WCSTEM=AQ(NORAW+3)*UC(NORAW+3)
C       WCH2O=AQ(NORAW+4)*UC(NORAW+4)
C       WCELEC=AQ(NORAW+5)*UC(NORAW+5)
C       WCFUEL=AQ(NORAW+6)*UC(NORAW+6)
C       WCANAL=AQ(NORAW+7)*UC(NORAW+7)
C       WCCONV=WCLAB+WCSTEM+WCH2O+WCELEC+WCFUEL+WCMNT+WCANAL+WCLF
C       WCPOVR=PCTOVR*(WCLAB+WCMNT+WCANAL+WCLF)/100.
C       WCADMO=PCTADM*WCLAB/100.
C       WORCAP(8)=WCFA*(WCLS+WCLIM+WCMGO+WCACID)+WCFB*(WCLS+WCCONV+WCPOVR
C      1 +WCADMO+WCLIM+WCMGO+WCACID)
C       IF(IECON.EQ.1) WORCAP(8) = WCFA*(WCLS+WCLIM+WCMGO+WCACID) +
C      1 WCFB * (WCCONV + WCPOVR) + WCFC * DIRECT(8) * 1000.
C       WORCAP(8)=WORCAP(8)/1000.
C       DO 1010 I=1,7
C       WORCAP(I)=WORCAP(8)*(DIRECT(I)/DIRECT(8))
C 1010 CONTINUE
       RETURN
       END
```

```
C
      SUBROUTINE SCRMOD(I)
C
C     SCRMOD - SCREEN ATTRIBUTES MODIFICATION PROGRAM
C        BY JEFF WINGET        8/20/85
C
C     I  -  FOLLOWING OPTIONS ARE AVAILABLE
C           ---------------------------------
C            0 - ALL ATTRIBUTES OFF
C            1 - BOLD ON (HIGH INTENSITY)
C            4 - UNDERSCORE ON (MONO ONLY)
C            5 - BLINK ON
C            7 - REVERSE VIDEO ON
C            8 - INVISIBLE PRINT
C
C           COLOR ATTRIBUTES
C
C            30 - BLACK FOREGROUND
C            31 - RED FOREGROUND
C            32 - GREEN FOREGROUND
C            33 - YELLOW FOREGROUND
C            34 - BLUE FOREGROUND
C            35 - MAGENTA FOREGROUND
C            36 - CYAN FOREGROUND
C            37 - WHITE FOREGROUND
C            40 - BLACK BACKGROUND
C            41 - RED BACKGROUND
C            42 - GREEN BACKGROUND
C            43 - YELLOW BACKGROUND
C            44 - BLUE BACKGROUND
C            45 - MAGENTA BACKGROUND
C            46 - CYAN BACKGROUND
C            47 - WHITE BACKGROUND
C
C
      CHARACTER*1 ESC
      DATA ESC/''/
      IF (I.LT.10) GOTO 10
      WRITE (6,100) ESC,I
  100 FORMAT (' ',A1,'[',I2,'m'\)
      GOTO 90
   10 WRITE (6,200) ESC,I
  200 FORMAT (' ',A1,'[',I1,'m'\)
   90 CONTINUE
      RETURN
      END
```

```
$DEBUG
$notstrict
C
      SUBROUTINE OUTTVA
C
C          THIS SUBROUTINE PRINTS OUT THE FOLLOWING OUTPUT SECTIONS:
C
C            - BOILER/SYSTEM PERFORMANCE
C            - SYSTEM MATERIAL BALANCE
C            - EMMISION SUMMARY
C            - CAPITAL COSTS
C            - ANNUAL COSTS
C
C          CERTAIN FINAL CLCULATIONS PERTINENT TO THE ABOVE ARE PERFORMED
C     IN THIS SUBROUTINE.
C
C          THE FOLLOWING OUTPUT SECTIONS ARE PRINTED IN THE INPUT AND
C     INDIVIDUAL MODULE SUBROUTINES RESPECTIVELY.
C
C            - INPUT SUMMARY
C            - MODULE SPECIFIC INFORMATION
C
C         REPORTS ARE WRITTEN TO THE TERMINAL AND/OR OFFLINE PRINTER
C     AS PER USER REQUEST.
C
C            WRITTEN BY PEDCO ENVIRONMENTAL INC. (JP)
C                  (5-12-83)
C
C
      CHARACTER*4 HMOD,HSTRM,HCAP,SHCAP,MONTH*9
      CHARACTER ATIT*25,UNITS*6
      INTEGER FMB,CT,BOTTOM,TYPCT,CLNLVL,FMAAHX,TADC,TCC,OVHDCB
      LOGICAL KO(2),TVA
      REAL INDCST,CAPCST
      REAL*8 WCC,SFF,AIR,ATD,LATD,LITC,LIT,IPT,LACC,LF,KL
      DIMENSION HMOD(10),HSTRM(5,8),NST(10),HCAP(10,10),SHCAP(6,8,10),
     &IC(10),CE(3),MONTH(12),ATIT(35),UNITS(35),RATE(35),AQ(35)
      INTEGER CCLEV
      INTEGER AP(35),PP(35)
      INTEGER R100,AC(35),TDIR,CC(10,8),TOTALA,OVHD,OVHDC,FYAC,TALEV,
     &        TIND,TCR,CAPCHG,ARLEV,TFI
      COMMON /OUTH/ HMOD,HSTRM,HCAP,SHCAP,IC,NST
      COMMON /COSTS/ CAPCST(10,8),ANN(25),INDCST(10)
      COMMON /COAL1/ HHV,PS,PA,CST,PNA,ALK,CL,FEO,PM
      COMMON /STREAM/ PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
      COMMON /INPUT/BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,HHVCLN,PSCLN,PACLN,CSTCLN,PNACLN,ALKCLN,
     &CLCLN,FEOCLN,PMCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,IAPCS(9),KO
     &,TVA
      COMMON /INSYSP/ CCEFF,UTEFF,BNHR,FANLOS,QIN,ALKIN
      COMMON /PARM/ P(500)
      COMMON /MATBAL/ TABLE(8,8)
```

```
      COMMON /FAN/ DELTAP,CSTFAN
      COMMON /ESC/ IYC,IMC,IYA,IMA,RATPLT,RATLAB,RATMAT,RATOM
      DATA MONTH/'  JANUARY',' FEBRUARY','     MARCH','     APRIL',
     &           '       MAY','      JUNE','      JULY','    AUGUST',
     &           'SEPTEMBER','  OCTOBER',' NOVEMBER',' DECEMBER'/
C
CC    ANNUAL COST TITLES
C
      DATA ATIT/'CALCITIC LIMESTONE        ','DOLOMITIC LIMESTONE       ',
     &          'CALCITIC HYDRATE (LIME)   ','DOLOMITIC HYDRATE (LIME) ',
     &          'CALCITIC PRESSURE HYDRATE','DOLOMITIC PRESSURE HYDRTE',
     &          'NAHCOLITE                 ','                          ',
     &          'MgO                       ','ADIPIC ACID               ',
     &          '                          ','                          ',
     &          '                   SYSTEM','SOLIDS DISPOSAL FACILITY',
     &          '                 ANALYSIS','SOLIDS DISPOSAL, WET     ',
     &          'SOLIDS DISPOSAL, DRY      ','                          ',
     &          'PROCESS WATER             ','ELECTRICITY               ',
     &          'STEAM                     ','                          ',
     &          'MAINTENANCE LABOR & MTRL.','                          ',
     &          '                          ','DIESEL FUEL               ',
     &          'FUEL COST PREMIUM         ','                          ',
     &          '                          ','                          ',
     &          '                          ','                          ',
     &          'LIMESTONE                 ','LIME                      ',
     &          '                          '/
C
CC    ANNUAL COST QUANTIY UNITS
C
      DATA UNITS/'TONS  ','TONS  ','TONS  ','TONS  ','TONS  ','TONS  ',
     &           'TONS  ','      ','TONS  ','TONS  ','      ','      ',
     &           'MANHRS','MANHRS','MANHRS','TONS  ','TONS  ','      ',
     &           'K GAL ','KWH   ','K LBS ','      ','   $  ','      ',
     &           '      ','GAL   ','TONS  ','      ','      ','      ',
     &           '      ','      ','TONS  ','TONS  ','      '/
C
CC    POINTER TO IAPCS ANNUAL COST QUANTITIES
C
      DATA AP/ 7,14, 8,15,16,17, 9, 0,18,19,22,23, 1, 3,21,12,13, 0,
     &         5, 4, 6, 0,10, 0, 0,20,11, 0, 0, 0, 0, 0, 0, 0, 0/
C
CC    POINTER TO P ARRAY
C
      DATA PP/156,163,157,164,168,169,158,  0,382,383,  0,  0,150,
     &        151,152,161,162,  0,154,153,155,  0,  0,  0,  0,389,
     &        160,  0,  0,  0,  0,  0,197,198,  0/
C
CC    STATEMENT FUNCTION TO ROUND TO NEAREST $100.
C
      R100(X)= IFIX(X*.01+.5)*100
      DO 3 I=1,10
      NST(I)=0
```

```
      IF(I.EQ.5 .OR. I.EQ.6) NST(I)=2
      IF(I.EQ.7) NST(I)= 3
      IF(I.EQ.8) NST(I)= 7
    3 CONTINUE
C
      WRITE(8,*) 'ANN ',ANN
      WRITE(8,*) ' INDCST ',INDCST
      WRITE(8,*) ' IC  ',IC
C
C     FINAL SYSTEM PERFORMANCE CALCULATIONS
C
C>>>> COAL USAGE
C>>>  TONS/H
      CUSE= QIN*1.E6/(HHV*2000.)
C>>>  ANNUAL (TONS/YR)
      ANCUSE= CUSE*8760.*CF
C>>>> IAPCS ENERGY PENALTY (BTU/KWH) - SUM OF ALL ENERGY USES
      TENGY= ANN(4)*3413./(BSIZE*1000.*8760.*CF)+ANN(6)/(BSIZE*1000.)
     &       +CCEFF
C>>>> SYSTEM NET GENRATION
      SNGEN= BSIZE*BNHR/(BNHR+TENGY)
C>>>> EFFICIENCY TO %
      UTEFF=UTEFF*100.
C
C     EMISSION SUMMARY CALCULATIONS
C
C>>>> COMPUTE LASTE ENTRY IN MATERIAL BALANCE TABLE
C>>>  FIND FIRST MODULE AFTER AIR HEATER EXIT
      DO 5 FMAAHX=1,3
      IF(IAPCS(FMAAHX) .GT. 3) GO TO 7
    5 CONTINUE
      FMAAHX=4
C>>   FIND NUMBER OF MODULES AFTER AIR HEATER EXIT
    7 NMAAHX= NM-FMAAHX+1
      LE= NMAAHX+2
C>>>> PERCENT REDUCTION
C>>>  PARTICULATE
      PPR= (TABLE(1,6)-TABLE(LE,6))/TABLE(1,6)*100.
      IF(PPR .GT. 99.9) PPR=99.9
C>>>  SO2
      PSR= (TABLE(1,7)-TABLE(LE,7))/TABLE(1,7)*100.
      IF(PSR .GT. 99.9) PSR=99.9
C>>>  NO2
      PNR= (TABLE(1,8)-TABLE(LE,8))/TABLE(1,8)*100.
      IF(PNR .GT. 99.9) PNR=99.9
C>>>> EMISSION RATES (LB/MMBTU)
C>>>  PARTICULATE
      ERPRT= PPHPRT/QIN
C>>>  SO2
      ERSO2= PPHSO2/QIN
C>>>  NO2
      ERNOX= PPHNOX/QIN
```

```
C>>>> CONCENTRATIONS (PPM)
C>>>  PARTICULATE
      PPMPRT= PPHPRT/PPH*1.E6
C>>>  SO2
      PPMSO2= PPHSO2/(ACFM*60.)*359.*(TEMP+460.)/520./64.*1.E6
C>>>  NO2
      PPMNOX= PPHNOX/(ACFM*60.)*359.*(TEMP+460.)/520./30.*1.E6
      WRITE(8,*) 'PPR  PSR  PNR  ERPRT  ERSO2  ERNOX  PPMPRT
     &PPMSO2  PPMNOX'
      WRITE(8,*) PPR,PSR,PNR,ERPRT,ERSO2,ERNOX,PPMPRT,PPMSO2,PPMNOX
C
C     SUM AND ROUND DIRECT CAPITAL COSTS
C
      TDIR=0
      DO 50 I=1,10
      DO 45 J=1,8
      WRITE(8,*) ' CAPCST (',I,',',J,')',CAPCST(I,J)
      CC(I,J)= R100(CAPCST(I,J))
C  45 CC(I,J)= R100(CAPCST(I,J))
   45 WRITE(8,*) ' CC ',CC(I,J)
      TDIR=TDIR+CC(I,1)
      WRITE(8,*) ' TDIR  ',TDIR
   50 CONTINUE
      TDIR= TDIR+R100(CSTFAN)
      TFI= TDIR
C
C     ANNUAL COST CALCULATIONS
C
C>>>> CALCULATE ANNUAL COSTS FROM QUANTITIES AND UNIT COSTS
C>>>  CALCULATE ANNUAL FUEL COST PREMIUM
C>>   SET ANNUAL CLEAN COAL QUANTITY
      IF(CLNLVL .EQ. 2) ANN(11)= ANCUSE
C>>   SET COAL CLEANING COST
      P(160)= CSTCLN-CSTROM
C>>>  COMPUTE ANNUAL COSTS
      DO 10 I=1,35
C>>     RATE CALCULATION
        RATE(I)= 0.0
        IF(I .EQ. 23) THEN
C>      MAINTENANCE LABOR AND MATERIALS
          RATE(I)= 1.0
        ELSEIF(PP(I) .NE. 0) THEN
C>      PP POINTS TO APPROPRIATE RATE IN THE PARM FILE
          RATE(I)= P(PP(I))*RATOM
        ENDIF
   10 CONTINUE
C>>     ANNUAL COST CALCULATION
      DO 15 I=1,35
        AC(I)= 0.0
        AQ(I)= 0.0
        IF(AP(I) .NE. 0) AQ(I)= ANN(AP(I))
        IF(RATE(I) .NE. 0.) AC(I)=R100(RATE(I)*AQ(I))
```

```
   15 CONTINUE
C>>    TOTALS: ANNUAL DIRECT, CONVERSION, OVERHEAD, TOTAL ANNUAL
       CONS= 0.0
       TOTALA= 0
       TADC= 0
       TCC= 0
       DO 14 I=1,35
         IF(I .GT. 11) TCC= TCC+AC(I)
   14    TADC= TADC + AC(I)
       WRITE(8,*) ' TCC TADC ',TCC,TADC
C>>>   COMPUTE ANNUAL COSTS
       OVHDCB= AC(13)+AC(14)+AC(15)+AC(23)
       OVHDC=R100(P(165)*OVHDCB*.01)
       TOTALA= TADC+OVHDC
       WRITE(8,*) ' OVHDC  TCC  TOTALA RATOM ',OVHDC,TCC,
      &TOTALA,RATOM
C
C      SUM INDIRECT CAPITAL COSTS
C
C      COMPUTE TOTAL FIXED INVESTMENT
C
       TIND= 0
       DO 17 I=1,5
       IC(I)= R100(INDCST(I))
       TIND= TIND+IC(I)
       WRITE(8,*) ' I,INDCST,TIND',I,INDCST(I),TIND
   17  TFI= TFI+IC(I)
       INDCST(7)= (TADC-TCC)/12.+TCC/9.+OVHD/9.+.03*TDIR
       DO 20 I= 6,10
       IC(I)= R100(INDCST(I))
       TIND= TIND+IC(I)
   20  CONTINUE
       TCR= TIND+TDIR
C>>>   COMPUTE LEVELIZED ANNUAL CAPITAL CHARGES
C>>    WEIGHTED COST OF CAPITAL
       IF (P(180) .EQ. 0.0) THEN
          WCC= .01*(P(182)*P(183)+P(184)*P(185)+P(186)*P(187))
       ELSE
          WCC= .01*P(180)
       ENDIF
       SFF= WCC/((1+WCC)**P(188)-1.)
       AIR=( -.0106*P(188)+.88)*.01
       ATD= 2*(WCC+SFF)*(P(189)-(1./(WCC+SFF)))/(P(189)*(P(189)+1.)
      &      *WCC)
       LATD= -(ATD-1./P(190))*P(191)/(1.-P(191))
       LITC= -.01*(WCC+SFF)*P(192)/((1.+WCC)*(1.-P(191)))
       LIT= ((WCC+SFF)+AIR-1./P(190))*(1.-P(182)*P(183)*.01/WCC)
      &      *P(191)/(1.-P(191))
       IPT= P(193)
       IF (P(166) .EQ. 0.) THEN
          LACC= WCC+SFF+AIR+LATD+LITC+LIT+IPT
       ELSE
```

```
         LACC= P(166)*.01
       ENDIF
       CCLEV= R100(TCR*SNGL(LACC))
       WRITE(8,*) ' WCC SFF AIR ATD LATD LITC LIT IPT LACC CCLEV '
       WRITE(8,*) WCC,SFF,AIR,ATD,LATD,LITC,LIT,IPT,LACC,CCLEV
C>>>   COMPUTE FIRST YEAR ANNUALIZED COST
       BCP= CCLEV+TOTALA
C>>>   CALCULATE LEVELIZED FIRST YEAR O&M
       KL= (1.+P(174)/100.)/(1.+P(196))
       IF(P(167) .EQ. 0.) THEN
          IF (KL.EQ.1.) THEN
             LF=WCC+SFF
           ELSE
            LF= (WCC+SFF)*KL*(1.-KL**P(190))/(1.-KL)
          ENDIF
       ELSE
         LF= P(167)
       ENDIF
       WRITE(8,*) ' KL  LF ',KL,LF
       TALEV= R100(SNGL(LF)*TOTALA)
C>>>   CALCULATE $/KW
       DPKW= TCR/(1000.*BSIZE)
C>>>   CALCULATE LEVELIZED ANNUAL REQUIREMENTS
       ARLEV= TALEV+CCLEV
C>>>   CALCULATE LEVELIZED ANNUAL REVENUE REQUIREMENTS
       RRLV= ARLEV/(BSIZE*8760.*CF)
C>>>   CALCULATE FIRST YEAR ANNUALIZED COST
       FYAC= TOTALA+CCLEV
C>>    FIRST YEAR REVENUE REQUIREMENT
       FYRR= FYAC/(BSIZE*8760.*CF)
       WRITE(8,*) ' BCP KL LF TALEV ARLEV RRLV FYAC FYRR '
       WRITE(8,*) BCP,KL,LF,TALEV,ARLEV,RRLV,FYAC,FYRR
C>>>   CALCULATE COST EFFECTIVENESS
       DO 30 I=6,8
       CE(I-5)= 0.
       DIF= TABLE(1,I)-TABLE(LE,I)
       IF (DIF .EQ. 0.) GOTO 30
       CE(I-5)= ARLEV/(DIF/2000.*8760.*CF)
       IF(CE(I-5) .LT. 0.) CE(I-5)= 0.
    30 CONTINUE
C
C      LOOP ON OUTPUT DEVICE
C
       DO 100 LD=6,7
       IF(.NOT. KO(LD-5)) GO TO 100
C
C      FINAL SYSTEM PERFORMANCE REPORT
C
C>>>> PRINT TITLE
       WRITE(LD,1000)
       WRITE(LD,1010)
C>>>   BODY OF REPORT
```

```
      WRITE(LD,1020) UTEFF,BNHR,QIN,CUSE,ANCUSE,TENGY,SNGEN
C
C     SYSTEM MATERIAL BALANCE REPORT
C
C>>>> PRINT TITLE
      WRITE(LD,1005)
      WRITE(LD,1030)
C>>>> PRINT COLUMN HEADINGS
      WRITE(LD,1040)
      IF(NMAAHX .NE. 0) WRITE(LD,1050)(HMOD(IAPCS(I)),
     &                  I=FMAAHX,NM)
      WRITE(LD,1060)
C>>>  PRINT BODY OF TABLE
      DO 35 I=1,8
C>>   'LE' IS LAST ENTRY IN TABLE
      IF(I.GT.2)WRITE(LD,1070)(HSTRM(K,I),K=1,5),(TABLE(J,I),J=1,LE)
   35 IF(I.LE.2)WRITE(LD,1075)(HSTRM(K,I),K=1,5),(TABLE(J,I),J=1,LE)
C
C     EMISSIONS SUMMARY REPORT
C
C>>>> PRINT TITLE
      WRITE(LD,1005)
      WRITE(LD,1080)
C>>>> PRINT COLUMN HEADINGS
      WRITE(LD,1090)
C>>>> CALCULATE % REDUCTION (PRT,SO2,NOX)
      WRITE(LD,1100) PPHPRT,PPR,ERPRT,PPHSO2,PSR,ERSO2,
     &               PPMSO2,PPHNOX,PNR,ERNOX,PPMNOX
C
C     CAPITAL COSTS REPORT
C
C>>>> PRINT TITLE
      WRITE(LD,1000)
      WRITE(LD,1110) MONTH(IMC),IYC
C>>>> PRINT BODY OF REPORT (LOOP ON CONTROL MODULE)
      DO 40 I=1,10
C>>>  PRINT CAP COSTS ONLY IF NOT = 0.0
      IF(CAPCST(I,1) .EQ. 0.0) GO TO 40
C>>>  CAPITAL COST FOR MODULE
      WRITE(LD,1120) (HCAP(J,I),J=1,10),CC(I,1)
C>>>  SUBTOTALS FOR MODULE (IF ANY)
      WRITE(8,*) ' NST= ',NST(I),I
      IF(NST(I) .EQ. 0 .OR. I .EQ. 5 .AND. CC(I,2).EQ.0) GO TO 40
      KNST= NST(I)
      WRITE(LD,1130) ((SHCAP(J,K,I),J=1,6),CC(I,K+1),K=1,KNST)
      WRITE(LD,1140)
   40 CONTINUE
C>>>  PRINT FAN COST
      IFC= R100(CSTFAN)
      IF(CSTFAN .NE. 0.0) WRITE(LD,1160) IFC
C>>> PRINT TOTAL
      WRITE(LD,1150) TDIR
```

```
C>>>> PRINT INDIRECTS
C>>>  PRINT TOTAL INDIRECTS
      WRITE(LD,1170) TIND
C>>>  PRINT SUBTOTALS FOR INDIRECTS
      WRITE(LD,1180) (IC(I),I=1,5),TFI,IC(9),IC(8),IC(6),IC(10),IC(7)
C>>>> PRINT TOTAL CAPITAL COST
      WRITE(LD,1190) TCR
C>>>  PRINT $/KWH
      WRITE(LD,1195) DPKW
C
C     ANNUAL COST REPORT
C
C>>>> PRINT REPORT HEADING
      WRITE(LD,1200) MONTH(IMA),IYA
C>>>> PRINT COLUMN HEADINGS
      WRITE(LD,1210)
C>>>> PRINT ANNUAL COSTS
C>>>  LOOP ON COST CATEGORIES
      DO 55 I=1,35
      IF(I .EQ. 13) WRITE(LD,1215)
      IF(I .EQ. 19) WRITE(LD,1217)
      IF(I .EQ. 23) THEN
        WRITE(LD,1218) ATIT(I),AC(I)
      ELSEIF(AQ(I) .NE. 0.) THEN
        WRITE(LD,1220) ATIT(I),AQ(I),UNITS(I),RATE(I),AC(I)
      ENDIF
   55 CONTINUE
C>>>  PRINT TOTAL CONVERSION AND DIRECT COSTS
      WRITE(LD,1225) TCC,TADC
C>>>  PRINT PLANT OVERHEAD
      WRITE(LD,1230) OVHDCB,P(165),OVHDC
C>>>  PRINT BUSBAR COST OF POWER
      WRITE(LD,1240) TOTALA,TCR,LACC,CCLEV,FYAC
C>>>  PRINT LEVELIZED FIRST YEAR O&M & ANNUAL REQUIREMENTS
      WRITE(LD,1250) TOTALA,LF,TALEV,TCR,LACC,CCLEV,ARLEV
C>>>  PRINT LEVELIZED ANNUAL REVENUE REQUIREMENTS
      WRITE(LD,1260) FYRR,RRLV
C>>>  PRINT COST EFFECTIVENESS
      WRITE(LD,1270) (CE(III),III=1,3)
  100 CONTINUE
      RETURN
C
C     OUTPUT FORMATS
C
 1000 FORMAT(1H1///)
 1005 FORMAT(//)
 1010 FORMAT(T28,'BOILER/SYSTEM PERFORMANCE',/T27,27('-'))
 1020 FORMAT(1H0,T11,'UNIT THERMAL EFFICIENCY',18(1H.),F8.1,1H%,
     &         /,T11,'BOILER NET HEAT RATE',21(1H.),F8.1,' BTU/KWH',
     &        //,T11,'HEAT INPUT',31(1H.),F8.1,' MMBTU/H',
     &         /,T11,'COAL USE',33(1H.),F8.1,' TONS/H',
     &        /,T11,'ANNUAL COAL CONSUMPTION',18(1H.),1PG12.4,' TONS/YR',
```

```
     &         //,T11,'IAPCS ENERGY PENALTY',21(1H.),0PF8.1,' BTU/KWH',
     &          /,T11,'SYSTEM NET GENERATION',20(1H.),F8.1,' MW')
 1030 FORMAT(///T29,'SYSTEM MATERIAL BALANCE',
     &        /T28,'(100% CAPACITY CONDITION)'/T27,27(1H-))
 1040 FORMAT(//T36,'AIR',/T26,'UNCONT- HEATER',/T26,'ROLLED    EXIT')
 1050 FORMAT(1H+,T43,5(A4,4X))
 1060 FORMAT(T25,56(1H-))
 1070 FORMAT(2X,5A4,' :',7F8.0)
 1075 FORMAT(2X,5A4,' :',-3P7F8.0)
 1080 FORMAT(T34,'EMISSION SUMMARY',
     &       /T29,27(1H-))
 1090 FORMAT(//T41,'PERCENT',/T16,'POLLUTANT',T31,'LB/HR',T40,
     &        'REDUCTION',T53,'LB/MMBTU',T65,'PPM(V)',/T15,11(1H-),T31,
     &        '-----',T40,9(1H-),T53,8(1H-),T65,'------')
 1100 FORMAT(T15,'PARTICULATE',F11.0,T42,F4.1,T50,F9.3,T64,
     &       /T19,'SO2',4X,F11.0,T42,F4.1,T50,F9.3,T64,F6.0,
     &        /T19,'NO2',4X,F11.0,T42,F4.1,T50,F9.3,T64,F6.0)
 1110 FORMAT(T19,'INSTALLED CAPITAL COSTS ',A9,',  ',I4,
     &       /T18,41(1H-))
 1120 FORMAT(//T11,10A4,3H--$,I10)
 1130 FORMAT(/T14,6A4,3H--$,I10)
 1140 FORMAT(/T11,43(1H-))
 1150 FORMAT(//T6,'TOTAL DIRECT CAPITAL COSTS',21(1H>),1H$,I10)
 1160 FORMAT(//T11,'FANS',38('-'),1H$,I10)
 1170 FORMAT(//T11,'INDIRECT COSTS',28(1H-),1H$,I10)
 1180 FORMAT(/T16,'ENGINEERING DESIGN & SPRV.$',I10,
     &        /T16,'ARCHITECT AND ENGINEERING.$',I10,
     &        /T16,'CONSTRUCTION EXPENSE.....$',I10,
     &        /T16,'CONTRACTOR FEES.....$',I10,
     &        /T16,'CONTINGENCY.....$',I10,
     &       //T11,'TOTAL FIXED INVESTMENT.....$',I10,
     &       //T16,'ALLOWANCE FOR STARTUP&MOD.$',I10,
     &        /T16,'INTEREST DURING CONSTRUC..$',I10,
     &        /T16,'ROYALTIES.....$',I10,
     &        /T16,'LAND.....$',I10,
     &        /T16,'WORKING CAPITAL.....$',I10)
 1190 FORMAT(//T51,13(1H*),/T6,'TOTAL CAPITAL ',30(1H>),3H>*$,
     &              I10,1H*,/T51,13(1H*))
 1195 FORMAT(/T6,'TOTAL SYSTEM COST',29(1H>),1H$,F8.2,'/KW'//)
 1200 FORMAT('1',T22,'ANNUAL OPERATING COSTS ',A9,',  ',I4,/T21
     &        ,40('-'))
 1210 FORMAT(//'  ITEM',T33,'QUANTITY',T52,'RATE',T67,'ANNUAL COST',
     &        /' ======',T32,'==========',T51,'======',T66,13(1H=),
     &//' DIRECT COSTS - FIRST YEAR',/' ========================='//
     &' RAW MATERIALS'/'+_____________')
 1215 FORMAT(/' CONVERSION COSTS',/' ----------------'//
     & ' OPERATING LABOR AND SUPERVISION'/
     & '+_____________________________')
 1217 FORMAT(/' UTILITIES'/'+_________')
 1218 FORMAT(1H A25,T62,1H$,I12)
 1220 FORMAT(1X,A25,T32,G9.4,1X,A6,T50,F7.2,T62,1H$,I12)
 1225 FORMAT(//'   TOTAL CONVERSION COST',T32,'  ',T51,T62,1H$,I12,
```

```
     &//'   TOTAL DIRECT COST',T32,'    ',T51,T62,1H$,I12))
 1230 FORMAT(//' PLANT OVERHEAD',T32,I9,' $',T51,F6.1,'%',T62,1H$,
     &        I12)
 1240 FORMAT(/' TOTAL FIRST YEAR O&M EXPENSE',T62,1H$,I12,
     &       /' LEVELIZED CAPITAL CHARGES',T28,I13,' $',T51,2PF6.1,1H%,
     &        T62,1H$,I12,//,' FIRST YEAR ANNUALIZED COST',T62,1H$,
     &        I12)
 1250 FORMAT(/' LEVELIZED FIRST YEAR O&M',T28,I13,' $',T51,F6.3,T62,
     &        1H$,I12,/' LEVELIZED CAPITAL CHARGES',T28,I13,' $',T51
     &       ,2PF6.1,1H%,T62,1H$,I12,//' LEVELIZED ANNUAL REQUIREMENTS',
     &        T62,1H$,I12)
 1260 FORMAT(///' FIRST YEAR ANNUAL REVENUE REQUIREMENTS',T55,F14.2,
     &          ' MILLS/KWH',/' LEVELIZED ANNUAL REVENUE REQUIREMENTS',
     &         T55,F14.2,' MILLS/KWH')
 1270 FORMAT(///' PARTICULATE COST EFFECTIVENESS',T45,F14.2,'  $/TON',
     &         /'          SO2 COST EFFECTIVENESS',T45,F14.2,'  $/TON',
     &         /'          NOX COST EFFECTIVENESS',T45,F14.2,'  $/TON')
C
      END
```

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     SUBPROGRAM TO ZERO VARIABLES
C     REVISION DATE 83-04-01
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE ZERO
C
      COMMON /PARM/ P(500)
      COMMON/BIG1/BAGDLP,BAGRAT,BAGCOST,BGLIFE,PSSO2X,EBYPAS,RS,PSC,
     1 CCSA,EFFPS,ESPDLP,RESIST,SCARAT,FANHP,FANLC,FANMC,DELTAP,FILTSA,
     2 PSF,FILRAT,H,TL,VD,VE,TGASID,SKLAND,DRAINM,DRAINL,CLAYSK,WPSO3,
     3 WPSO4,WPCO3,WPINS,ASHCLN,WINDEX,PDMAX,FLOOD,SRAIR,OXAIR,OXSCFM,
     4 PRES,SO2RT,SO2RV,SO2RST,SDFEE,AVPD,VT,HVCLN,HPTONW,
     5 ISLUDG,IFIXS,JSR,IVPD,IT,LON,LGMAX,ISTAND,NSBR,
     6 KEPASS,KPASO2,ICLAR,JSSO,JDIS,JSTR,JGPM,IRH,NSBF
      COMMON/COAL/HVC,WPC,WPH,WPO,WPN,WPSUL,WPCL,WPASH,WPH2O,SULO,ASHO
      COMMON/COMPLE/ICOMPL,IRPT,IRUN,NEXT,IEQPR,IWTBAL,ITM,INPOPT,NOPART
      COMMON/FLYASH/IASH,ASHUPS,ASHSCR,ASHPRU,ASHPRS,ASHLBU,ASHLBS
      COMMON/HGAS2/MPCT,LBMH,LBH,SCFM,HHG,HWG,CFR,BCFR,RCFR
      REAL MPCT(6),LBMH(6),LBH(6)
      COMMON/LANDIS/FILAND,TWLF,DISFUL,ELECLF,WATERF,FILANL,OVLF
      REAL LGMAX
      COMMON/OPPSCH/IOPSCH,IYROP,IA(50)
      COMMON/PCOST/POND(9),PDACRE,TOPHRS,PNDCAP,PNDLAB(9),PNDMAT(9),ELAN
      COMMON/PND1/PDEPTH,PMXEXC,DISTPD,ILINER,CLAYIN,CLAYUC,HYPLMC,
     1            HYPLLC,PSAMAX,ACREC,APRAIN,SAPILE
      COMMON/RRIND/XINT,PCTMNT,PDMNTP,XINFLA,PCTOVR,PCTADM,UNDCAP,
     1             PCTINS,UC,IECON
      REAL *8 XINT,PCTMNT,PDMNTP,XINFLA,PCTOVR,PCTADM,UNDCAP,PCTINS,UC(1
     1       8)
      COMMON/SCRUBR/LG,V,PPMOUT,SO2R,ELIMIT,PR,PPD,WPS,OX,PH,NSTAGE,
     1              NGRID,HS,ACS,ACSTRN
      REAL LG
      COMMON/SOLDIS/CASO3,CASO4,CACO3,INSOLS,WDIS,PSD,DENS,DEND
      REAL INSOLS
      COMMON/PCOSTS/PCLEAR,PDEXEC,SODM,SODL,PERIML,PERIMM,ROADMM,ROADLL,
     1 RECLAM
      COMMON/PNDARW/AB,AID,AIE,AOD,AS,DL,P1,VEC,VETT,WELLM,WELLL,WPSITE
      COMMON/PNDIND/INDPND,PENGIN,PARCH,PFLDEX,PFEES,PCONT,PSTART,PCONIN
      COMMON/TAXES/ITAXFR,TXFRRT,TXRAT,FRRAT,SERVRT,ROYALT,IOTIME,OTRATE
      COMMON/CSTNDX/ICEPYR,CEPNDX,INVYR,IREVYR,RATLAB,RATMAT,ARM,ARL
      COMMON /TVAFAN/ TGAS,TWG,VPD,NTRAIN,DLTAPF,NOTRAN,NTOTAL,ISCRUB,
     &                NOREDN,PD,DPRH
C
      DO 1000 I=1,50
      IA(I)=0
 1000 CONTINUE
      DO 1010 I=1,18
      UC(I)=0.0
 1010 CONTINUE
```

```
      ITAXFR=P(373)
      TXFRRT=P(374)+P(375)
      RATLAB=263.9/323.4
      RATMAT=336.2/346.3
      TOPHRS=0.0
      WELLM=5000.
      WELLL=5100.
      RECLAM=3100.
      PCLEAR=1950.
      PDEXEC=1.75
      SODM=0.30
      SODL=0.49
      PERIML=6.05
      PERIMM=7.00
      ROADMM=6.00
      ROADLL=8.50
      CLAYUC = 0.0
      CLAYIN = 0.0
      HYPLMC = 0.0
      HYPLLC = 0.0
      DPRH=0.0
      ACREC=P(181)
      ASHO=80
      ASHPRU=0
      DISTPD=P(354)
      ILINER=P(355)
      IRPT=0
      ISLUDG=5
      OX=P(328)
      PARCH=P(397)
      PCONIN=P(364)
      PCONT=P(400)
      PDEPTH=P(352)
      PENGIN=P(396)
      PFEES=P(399)
      PFLDEX=P(398)
      PMXEXC=P(353)
      PNDCAP=P(391)
      PSD=P(324)
      PSTART=P(401)
      RCFR=0
      WDIS =0
      WPASH=0
      IFIXS=P(350)
C
      TGAS=0
      TWG=0
      VPD=0
      NTRAIN=0
      DLTAPF=0
      NOTRAN=0
      NTOTAL=0
```

```
ISCRUB=0
NOREDN=0
PD=0
RETURN
END
```

```
$DEBUG
$notstrict
C
      SUBROUTINE OUTEPR
C
C          THIS SUBROUTINE PRINTS OUT THE FOLLOWING OUTPUT SECTIONS:
C
C             - BOILER/SYSTEM PERFORMANCE
C             - SYSTEM MATERIAL BALANCE
C             - EMMISION SUMMARY
C             - CAPITAL COSTS
C             - ANNUAL COSTS
C
C          CERTAIN FINAL CLCULATIONS PERTINENT TO THE ABOVE ARE PERFORMED
C     IN THIS SUBROUTINE.
C
C          THE FOLLOWING OUTPUT SECTIONS ARE PRINTED IN THE INPUT AND
C     INDIVIDUAL MODULE SUBROUTINES RESPECTIVELY.
C
C             - INPUT SUMMARY
C             - MODULE SPECIFIC INFORMATION
C
C          REPORTS ARE WRITTEN TO THE TERMINAL AND/OR OFFLINE PRINTER
C     AS PER USER REQUEST.
C
C             WRITTEN BY PEDCO ENVIRONMENTAL INC. (JP)
C                    (5-12-83)
C
C
      CHARACTER*4 HMOD,HSTRM,HCAP,SHCAP
      CHARACTER*80 COMM1,COMM2,MONTH*9,ATIT*25,UNITS*6
      INTEGER FMB,CT,BOTTOM,TYPCT,CLNLVL,FMAAHX,TPC,TPI,TCR
      INTEGER AP(35),PP(35)
      LOGICAL KO(2),TVA
      REAL INDCST,CAPCST,LACC,OLMF
      DIMENSION HMOD(10),HSTRM(5,8),NST(10),HCAP(10,10),SHCAP(6,8,10),
     &IC(10),CE(3),MONTH(12),ATIT(35),UNITS(35),RATE(35),AQ(35)
      INTEGER R100,AC(35),TDIR,CC(10,8),TOTALA,OVHD,OVHDC,FYAC,TALEV,
     &        TIND,CAPCHG,ARLEV,CCLEV
      COMMON /OUTH/ HMOD,HSTRM,HCAP,SHCAP,IC,NST
      COMMON /COSTS/ CAPCST(10,8),ANN(25),INDCST(10)
      COMMON /COAL1/ HHV,PS,PA,CST,PNA,ALK,CL,FEO,PM
      COMMON /STREAM/ PPH,ACFM,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
      COMMON /INPUT/BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,HHVCLN,PSCLN,PACLN,CSTCLN,PNACLN,ALKCLN,
     &CLCLN,FEOCLN,PMCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,IAPCS(9),KO
     &,TVA
      COMMON /INSYSP/ CCEFF,UTEFF,BNHR,FANLOS,QIN,ALKIN
      COMMON /PARM/ P(500)
      COMMON /MATBAL/ TABLE(8,8)
      COMMON /FAN/ DELTAP,CSTFAN
      COMMON /ESC/ IYC,IMC,IYA,IMA,RATPLT,RATLAB,RATMAT,RATOM
```

```
      DATA MONTH/'  JANUARY',' FEBRUARY','    MARCH','    APRIL',
     &           '      MAY','     JUNE','     JULY','   AUGUST',
     &           'SEPTEMBER','  OCTOBER',' NOVEMBER',' DECEMBER'/
C
C     ANNUAL COST TITLES
C
      DATA ATIT/'  SYSTEM                    ','  WASTE DISPOSAL FACILITY',
     &          'ANALYSIS                    ','MAINTENANCE LABOR         ',
     &          'MAINTENANCE MATERIAL        ','ADMIN. & SUPPORT LABOR    ',
     &          '                            ','                          ',
     &          '                            ','FIXED COMPONENT           ',
     &          'VARIABLE COMPONENT          ','                          ',
     &          'SOLIDS DISPOSAL, WET        ','SOLIDS DISPOSAL, DRY      ',
     &          '                            ','CALCITIC LIMESTONE        ',
     &          'DOLOMITIC LIMESTONE         ','CALCITIC HYDRATE (LIME)   ',
     &          'DOLOMITIC HYDRATE (LIME)    ','CALCITIC PRESSURE HYDRATE',
     &          'DOLOMITIC PRESSURE HYDRTE   ','NAHCOLITE                 ',
     &          '                            ','MgO                       ',
     &          'ADIPIC ACID                 ','                          ',
     &          'WATER                       ','STEAM                     ',
     &          'ELECTRICITY                 ','                          ',
     &          'DIESEL FUEL                 ','FUEL COST PREMIUM         ',
     &          'LIMESTONE                   ','LIME                      ',
     &          '                            '/
C
C     ANNUAL COST QUANTIY UNITS
C
      DATA UNITS/'MANHRS','MANHRS','MANHRS','  $   ','  $   ','  $   ',
     &           '      ','      ','      ','  $   ','  $   ','      ',
     &           'TONS  ','TONS  ','      ','TONS  ','TONS  ','TONS  ',
     &           'TONS  ','TONS  ','TONS  ','TONS  ','      ','TONS  ',
     &           'TONS  ','      ','K GAL ','K LBS ','KWH   ','      ',
     &           'GAL   ','LBS   ','TONS  ','TONS  ','      '/
C
C     POINTER TO IAPCS ANNUAL COST QUANTITIES
C
      DATA AP/ 1, 3,21,10,10,  0,  0,  0,  0,  0,  0,  0,12,13,  0,  7,14,  8,
     &        15,16,17,  9,  0,18,19,  0,  5,  6,  4,  0,20,11,22,23,  0/
C
C     POINTER TO P ARRAY
C
      DATA PP/150,151,152,   0,   0,191,   0,   0,   0,   0,   0,   0,161,
     &        162,   0,156,182,157,183,184,185,158,   0,382,383,   0,
     &        154,155,153,   0,389,   0,186,187,   0/
C
C     STATEMENT FUNCTION TO ROUND TO NEAREST $100.
C
      R100(X)= IFIX(X*.01+.5)*100
C
      WRITE(8,*) '**** OUTEPR ****'
      WRITE(8,*)HCAP
      WRITE(8,*) 'ANN ',ANN
```

```
      WRITE(8,*) ' INDCST ',INDCST
      WRITE(8,*) ' IC   ',IC
      WRITE(8,*) ' ACFM ',ACFM
      WRITE(8,*) '  AC   ',AC
      DO 7777 JJJ=1,35
 7777 AC(JJJ) = 0
      DO 3 I=1,10
      NST(I)=0
      IF(I.EQ.5 .OR. I.EQ.6) NST(I)=2
      IF(I.EQ.7) NST(I)= 3
      IF(I.EQ.8) NST(I)= 7
    3 CONTINUE
C
C     FINAL SYSTEM PERFORMANCE CALCULATIONS
C
C>>>> COAL USAGE
C>>>  TONS/H
      CUSE= QIN*1.E6/(HHV*2000.)
C>>>  ANNUAL (TONS/YR)
      ANCUSE= CUSE*8760.*CF
C>>>> IAPCS ENERGY PENALTY (BTU/KWH) - SUM OF ALL ENERGY USES
      TENGY= ANN(4)*3413./(BSIZE*1000.*8760.*CF)+ANN(6)/(BSIZE*1000.)
     &       +CCEFF
C>>>> SYSTEM NET GENRATION
      SNGEN= BSIZE*BNHR/(BNHR+TENGY)
C>>>> EFFICIENCY TO %
      UTEFF=UTEFF*100.
C
C     EMISSION SUMMARY CALCULATIONS
C
C>>>> COMPUTE LASTE ENTRY IN MATERIAL BALANCE TABLE
C>>>  FIND FIRST MODULE AFTER AIR HEATER EXIT
      DO 5 FMAAHX=1,3
      IF(IAPCS(FMAAHX) .GT. 3) GO TO 7
    5 CONTINUE
      FMAAHX=4
C>>   FIND NUMBER OF MODULES AFTER AIR HEATER EXIT
    7 NMAAHX= NM-FMAAHX+1
      LE= NMAAHX+2
C>>>> PERCENT REDUCTION
C>>>  PARTICULATE
      PPR= (TABLE(1,6)-TABLE(LE,6))/TABLE(1,6)*100.
      TPYP= PPR*TABLE(1,6)*.01*8760.*CF
      IF(PPR .GT. 99.9) PPR=99.9
C>>>  SO2
      PSR= (TABLE(1,7)-TABLE(LE,7))/TABLE(1,7)*100.
      TPYS= PSR*TABLE(1,7)*.01*8760.*CF
      IF(PSR .GT. 99.9) PSR=99.9
C>>>  NO2
      PNR= (TABLE(1,8)-TABLE(LE,8))/TABLE(1,8)*100.
      TPYN= PNR*TABLE(1,8)*.01*8760.*CF
      IF(PNR .GT. 99.9) PNR=99.9
```

```
C>>>> EMISSION RATES (LB/MMBTU)
C>>>  PARTICULATE
      ERPRT= PPHPRT/QIN
C>>>  SO2
      ERSO2= PPHSO2/QIN
C>>>  NO2
      ERNOX= PPHNOX/QIN
C>>>> CONCENTRATIONS (PPM)
C>>>  PARTICULATE
      PPMPRT= PPHPRT/PPH*1.E6
C>>>  SO2
      PPMSO2= PPHSO2/(ACFM*60.)*359.*(TEMP+460.)/520./64.*1.E6
C>>>  NO2
      PPMNOX= PPHNOX/(ACFM*60.)*359.*(TEMP+460.)/520./30.*1.E6
      WRITE(8,*) 'PPR  PSR  PNR  ERPRT  ERSO2  ERNOX  PPMPRT
     &PPMSO2  PPMNOX'
      WRITE(8,*) PPR,PSR,PNR,ERPRT,ERSO2,ERNOX,PPMPRT,PPMSO2,PPMNOX
C
C     SUM AND ROUND DIRECT CAPITAL COSTS
C
      TDIR=0
      DO 50 I=1,10
      DO 45 J=1,8
      WRITE(8,*) ' CAPCST (',I,',',J,')',CAPCST(I,J)
      CC(I,J)= R100(CAPCST(I,J))
C  45 CC(I,J)= R100(CAPCST(I,J))
   45 WRITE(8,*) ' CC ',CC(I,J)
      TDIR=TDIR+CC(I,1)
      WRITE(8,*) ' TDIR  CSTFAN',TDIR,CSTFAN
   50 CONTINUE
      TDIR= TDIR+R100(CSTFAN)
C>>>> SET CONSTRUCTION PERIOD FACTOR
      DURAT= FLOAT(IYA)+(IMA-1)/12.-(FLOAT(IYC)+(IMC-1)/12.)
C>>>> PROCESS CAPITAL
      PCAP= TDIR
C>>>> COMPUTE TOTAL PLANT COST
      TPC= PCAP
      WRITE(8,*) 'PCAP  TPC  TPI  P(168) ',PCAP,TPC,TPI,P(168)
C
C     ANNUAL COST CALCULATIONS
C
C>>>> CALCULATE ANNUAL COSTS FROM QUANTITIES AND UNIT COSTS
C>>>  ASSIGN TOTAL DIRECT CAPITAL TO MAINTENANCE QUAINTITY
C>>>  CALCULATE ANNUAL FUEL COST PREMIUM
C>>   SET ANNUAL CLEAN COAL QUANTITY
      IF(CLNLVL .EQ. 2) ANN(11)= ANCUSE
C>>   SET COAL CLEANING COST
      P(160)= CSTCLN-CSTROM
C>>>  ESCALATE ANNUAL UNIT COSTS
C>>>  COMPUTE ANNUAL COSTS
      DO 10 I=1,35
C>>     RATE CALCULATION
```

```
        RATE(I)= 0.0
        IF(I .EQ. 4) THEN
          RATE(I)= .4
        ELSEIF(I .EQ. 5) THEN
          RATE(I)= .6
        ELSEIF(I .EQ. 6) THEN
          RATE(I)= P(PP(I))*.01
        ELSEIF(I .EQ. 10) THEN
          RATE(I)= CF
        ELSEIF(I .EQ. 11) THEN
          RATE(I)= 1.-CF
        ELSEIF(PP(I) .NE. 0) THEN
          RATE(I)= P(PP(I))*RATOM
        ENDIF
   10 CONTINUE
C>>     ANNUAL COST CALCULATION
      DO 15 I=1,35
        AQ(I)= 0.0
        AC(I)= 0.0
        IF(I .EQ. 6) THEN
C>        ADMINISTRATIVE AND SUPPORT LABOR
          AQ(6)= AC(1)+AC(2)+AC(3)+AC(4)
        ELSEIF(I .EQ. 10 .OR. I .EQ. 11) THEN
C>        FIXED AND VARIABLE COMPONENT
          AQ(I)=AC(1)+AC(2)+AC(3)+AC(4)+AC(5)+AC(6)
          WRITE(8,*) ' FV',(AC(J),J=1,6)
        ELSEIF(AP(I) .NE. 0) THEN
          AQ(I)= ANN(AP(I))
        ENDIF
        IF(RATE(I) .NE. 0.) AC(I)=R100(RATE(I)*AQ(I))
   15 CONTINUE
      CONS= 0.0
      TOTALA= 0
      DO 14 I=1,35
        IF(I .GT. 11) CONS= CONS+AC(I)
   14   IF(I .NE.10 .AND. I .NE.11)  TOTALA= TOTALA + AC(I)
      WRITE(8,*) ' CONS  TOTALA ',CONS,TOTALA
C
C     SUM INDIRECT CAPITAL COSTS
C
      TIND= 0
      DO 20 I=1,5
      IC(I)= R100(INDCST(I))
      TIND= TIND+IC(I)
   20 TPC= TPC+IC(I)
C>>>> COMPUTE CHARGE ON CAPITAL,AFDC, AND O & M LEVELIZATION FACTORS
      WRITE(8,*) ' CALLING SUBROUTINE EPECON'
      CALL EPECON(CCF1,OLMF,DURAT,AF)
      IF (P(163) .NE. 0.) OLMF= P(163)
      IF (P(164) .NE. 0.) CCF1= P(164)
      WRITE(8,*) 'TCR  IBL  CCF1  OLMF TCR P(163) P(164) AF',TCR,
     &IBL,CCF1,OLMF,TCR,P(163),P(164),AF
```

```
      TPI= R100(TPC*AF)
      TCR= TPI
C>>>> COMPUTE PRE-PRODUCTION COSTS
      IC(7)= R100(OMC/12. + CONS/(CF*12.) + .02*TPI)
C>>>> COMPUTE INVENTORY CAPITAL
      IC(8)= R100(CONS/(6.*CF))
      DO 22 I=6,10
      IF(I .NE. 7 .AND. I .NE. 8) IC(I)= R100(INDCST(I))
C>>>> TOTAL INDIRECTS
      TIND= TIND+IC(I)
C>>>> TOTAL CAPITAL REQUIREMENT
   22 TCR= TCR+IC(I)
      CAPCHG= R100(TCR*CCF1)
C>>   CALCULATE LEVELIZED 1ST YEAR O&M
      TALEV= R100(TOTALA*OLMF)
C>>   CALCULATE $/KW
      DPKW= TCR/(BSIZE*1000.)
C>>>> CALCULATE LEVELIZED ANNUAL REQUIREMENT
      ARLEV= TALEV+CAPCHG
C>>>  CALCULATE LEVELIZED ANNUAL REVENUE REQUIREMENTS
      RRLV= ARLEV/(BSIZE*8760.*CF)
C>>>  CALCULATE FIRST YEAR ANNUALIZED COST
      FYAC= TOTALA+CAPCHG
C>>   FIRST YEAR REVENUE REQUIREMENT
      FYRR= FYAC/(BSIZE*8760.*CF)
      WRITE(8,*) 'CAPCHG TALEV DPKW ARLEV RRLV FYAC FYRR'
      WRITE(8,*) CAPCHG,TALEV,DPKW,ARLEV,RRLV,FYAC,FYRR
C>>>  CALCULATE COST EFFECTIVENESS
      DO 30 I=6,8
      CE(I-5)= 0.
      DIF= TABLE(1,I)-TABLE(LE,I)
      IF (DIF .EQ. 0.) GOTO 30
      CE(I-5)= ARLEV/(DIF/2000.*8760.*CF)
      IF( CE(I-5) .LT. 0.) CE(I-5)= 0.
   30 CONTINUE
C
C     LOOP ON OUTPUT DEVICE
C
      DO 100 LD=6,7
      IF(.NOT. KO(LD-5)) GO TO 100
C
C     FINAL SYSTEM PERFORMANCE REPORT
C
C>>>> PRINT TITLE
      WRITE(LD,1000)
      WRITE(LD,1010)
C>>>  BODY OF REPORT
      WRITE(LD,1020) UTEFF,BNHR,QIN,CUSE,ANCUSE,TENGY,SNGEN
      WRITE(8,*) 'UTEFF,BNHR,QIN,CUSE,ANCUSE,TENGY,SNGEN'
      WRITE(8,*) UTEFF,BNHR,QIN,CUSE,ANCUSE,TENGY,SNGEN
C
C     SYSTEM MATERIAL BALANCE REPORT
```

```
C
C>>>> PRINT TITLE
      WRITE(LD,1005)
      WRITE(LD,1030)
C>>>> PRINT COLUMN HEADINGS
      WRITE(LD,1040)
      IF(NMAAHX .NE. 0) WRITE(LD,1050)(HMOD(IAPCS(I)),
     &                   I=FMAAHX,NM)
      WRITE(LD,1060)
C>>>  PRINT BODY OF TABLE
      DO 35 I=1,8
C>>   'LE' IS LAST ENTRY IN TABLE
      IF(I.GT.2)WRITE(LD,1070)(HSTRM(K,I),K=1,5),(TABLE(J,I),J=1,LE)
   35 IF(I.LE.2)WRITE(LD,1075)(HSTRM(K,I),K=1,5),(TABLE(J,I),J=1,LE)
C
C     EMISSIONS SUMMARY REPORT
C
C>>>> PRINT TITLE
      WRITE(LD,1005)
      WRITE(LD,1080)
C>>>> PRINT COLUMN HEADINGS
      WRITE(LD,1090)
C>>>> CALCULATE % REDUCTION (PRT,SO2,NOX)
      WRITE(LD,1100) PPHPRT,PPR,ERPRT,PPHSO2,PSR,ERSO2,
     &               PPMSO2,PPHNOX,PNR,ERNOX,PPMNOX
C
C     CAPITAL COSTS REPORT
C
C>>>> PRINT TITLE
      WRITE(LD,1000)
      WRITE(LD,1110) MONTH(IMC),IYC
C>>>> PRINT BODY OF REPORT (LOOP ON CONTROL MODULE)
      DO 40 I=1,10
C>>>  PRINT CAP COSTS ONLY IF NOT = 0.0
      IF(CAPCST(I,1) .EQ. 0.0) GO TO 40
C>>>  CAPITAL COST FOR MODULE
      WRITE(LD,1120) (HCAP(J,I),J=1,10),CC(I,1)
C>>>  SUBTOTALS FOR MODULE (IF ANY)
      IF(NST(I) .EQ. 0 .OR. I .EQ. 5 .AND. CC(I,2).EQ.0) GO TO 40
      KNST= NST(I)
      WRITE(LD,1130) ((SHCAP(J,K,I),J=1,6),CC(I,K+1),K=1,KNST)
      WRITE(LD,1140)
   40 CONTINUE
C>>>  PRINT FAN COST
      IFC= R100(CSTFAN)
      IF(CSTFAN .NE. 0.0) WRITE(LD,1160) IFC
C>>> PRINT TOTAL
      WRITE(LD,1150) TDIR
C>>>> PRINT INDIRECTS
C>>>  PRINT TOTAL INDIRECTS
      WRITE(LD,1170) TIND
C>>>  PRINT SUBTOTALS FOR INDIRECTS
```

```
      WRITE(LD,1180) (IC(I),I=1,5),TPC,TPI,(IC(I),I=6,10)
      WRITE(8,*) '(IC(I),I=1,5),TPC,TPI,(IC(I),I=6,10)'
      WRITE(8,*) (IC(I),I=1,5),TPC,TPI,(IC(I),I=6,10)
C>>>> PRINT TOTAL CAPITAL COST
      WRITE(LD,1190) TCR
C>>>  PRINT $/KWH
      WRITE(LD,1195) DPKW
C
C     ANNUAL COST REPORT
C
C>>>> PRINT REPORT HEADING
      WRITE(LD,1200) MONTH(IMA),IYA
C>>>> PRINT COLUMN HEADINGS
      WRITE(LD,1210)
      WRITE(LD,1212)
C>>>> PRINT ANNUAL COSTS
C>>>  LOOP ON COST CATEGORIES
      DO 80 I=1,35
        IF(I .EQ. 13) WRITE(LD,1215)
        IF(AQ(I) .NE. 0.0) WRITE(LD,1220)ATIT(I),AQ(I),UNITS(I),RATE(I),
     &                                          AC(I)
   80 CONTINUE
C>>> PRINT PLANT OVERHEAD
C     WRITE(LD,1230) OVHD,P(165),OVHDC
C>>>  PRINT FIRST YEAR ANNUALIZED COST
      WRITE(LD,1240) TOTALA,TCR,CCF1,CAPCHG,FYAC
C>>>  PRINT LEVELIZED FIRST YEAR O&M & ANNUAL REQUIREMENTS
      WRITE(LD,1250) TOTALA,OLMF,TALEV,TCR,CCF1,CAPCHG,ARLEV
C>>>  PRINT LEVELIZED ANNUAL REVENUE REQUIREMENTS
      WRITE(LD,1260) FYRR,RRLV
C>>>   PRINT COST EFFECTIVENESS
      WRITE(LD,1270) (CE(III),III=1,3)
  100 CONTINUE
C
C
      RETURN
C
C     OUTPUT FORMATS
C
 1000 FORMAT(1H1///)
 1005 FORMAT(//)
 1010 FORMAT(T28,'BOILER/SYSTEM PERFORMANCE',
     &       ,/T28,'(100% CAPACITY CONDITION)',/T27,27(1H-))
 1020 FORMAT(1H0,T11,'UNIT THERMAL EFFICIENCY',18(1H.),F8.1,1H%,
     &         /,T11,'BOILER NET HEAT RATE',21(1H.),F8.1,' BTU/KWH',
     &        //,T11,'HEAT INPUT',31(1H.),F8.1,' MMBTU/H',
     &         /,T11,'COAL USE',33(1H.),F8.1,' TONS/H',
     &        /,T11,'ANNUAL COAL CONSUMPTION',18(1H.),1PG12.4,' TONS/YR',
     &        //,T11,'IAPCS ENERGY PENALTY',21(1H.),0PF8.1,' BTU/KWH',
     &         /,T11,'SYSTEM NET GENERATION',20(1H.),F8.1,' MW')
 1030 FORMAT(///T29,'SYSTEM MATERIAL BALANCE'
     &       ,/T28,'(100% CAPACITY CONDITION)',/T27,27(1H-))
```

```
1040 FORMAT(//T36,'AIR',/T26,'UNCONT- HEATER',/T26,'ROLLED   EXIT')
1050 FORMAT(1H+,T43,5(A4,4X))
1060 FORMAT(T25,56(1H-))
1070 FORMAT(2X,5A4,' :',7F8.0)
1075 FORMAT(2X,5A4,' :',-3P7F8.0)
1080 FORMAT(T34,'EMISSION SUMMARY'
    &       /T29,27(1H-))
1090 FORMAT(//T41,'PERCENT',/T16,'POLLUTANT',T31,'LB/HR',T40,
    &        'REDUCTION',T53,'LB/MMBTU',T65,'PPM(V)',/T15,11(1H-),T31,
    &        '-----',T40,9(1H-),T53,8(1H-),T65,'------')
1100 FORMAT(T15,'PARTICULATE',F11.0,T42,F4.1,T50,F9.3,T64,
    &       /T19,'SO2',4X,F11.0,T42,F4.1,T50,F9.3,T64,F6.0,
    &        /T19,'NO2',4X,F11.0,T42,F4.1,T50,F9.3,T64,F6.0)
1110 FORMAT(T19,'INSTALLED CAPITAL COSTS  ',A9,',  ',I4,
    &       /T18,42(1H-))
1120 FORMAT(//T11,10A4,3H--$,I10)
1130 FORMAT(/T14,6A4,3H--$,I10)
1140 FORMAT(/T11,43(1H-))
1150 FORMAT(//T6,'TOTAL DIRECT CAPITAL COSTS',21(1H>),1H$,I10)
1160 FORMAT(//T11,'FANS',38('-'),1H$,I10)
1170 FORMAT(//T11,'INDIRECT COSTS',28(1H-),1H$,I10)
1180 FORMAT(/T16,'GENERAL FACILITIES.....$',I10,
    &        /T16,'ENGINEERING/HOME OFFICE...$',I10,
    &        /T16,'PROJECT CONTINGENCY.......$',I10,
    &        /T16,'PROCESS CONTINGENCY.......$',I10,
    &        /T16,'SALES TAX.................$',I10,
    &//T10,'TOTAL PLANT COST.................$',I10,
    & /T10,'TOTAL PLANT INVESTMENT..........$',I10,
    &       //T16,'ROYALTY ALLOWANCE.........$',I10,
    &        /T16,'PREPRODUCTION COSTS.......$',I10,
    &        /T16,'INVENTORY CAPITAL.........$',I10,
    &        /T16,'INITIAL CATALYST..........$',I10,
    &        /T16,'LAND......................$',I10)
1190 FORMAT(//T51,13(1H*),/T6,'TOTAL CAPITAL REQUIREMENT',19(1H>),
    &3H>*$,I10,1H*,/T51,13(1H*))
1195 FORMAT(/T6,'TOTAL SYSTEM COST',29(1H>),1H$,F8.2,'/KW'//)
1200 FORMAT('1',T22,'ANNUAL OPERATING COSTS  ',A9,',  ',I4,/T21,41('-'
    &       ))
1210 FORMAT(//'  ITEM',T33,'QUANTITY',T52,'RATE',T67,'ANNUAL COST',
    &        /' ======',T32,'==========',T51,'======',T66,13(1H=))
1212 FORMAT(/' OPERATING AND SUPERVISORY LABOR')
1215 FORMAT(//' CONSUMABLES',/'+_________________'/)
1220 FORMAT(1X,A25,T32,G9.4,1X,A6,T50,F7.2,T62,1H$,I12)
1230 FORMAT(/' PLANT OVERHEAD',T32,I9,' $',T51,F6.1,'%',T62,1H$,
    &        I12)
1240 FORMAT(/' TOTAL FIRST YEAR O&M EXPENSE',T62,1H$,I12,
    &       /' LEVELIZED CARRYING CHARGES',T28,I13,' $',T51,2PF6.1,1H%,
    &        T62,1H$,I12,//,' BUSBAR COST OF POWER',T62,1H$,
    &        I12)
1250 FORMAT(/' LEVELIZED FIRST YEAR O&M',T28,I13,' $',T51,F6.3,T62,
    &        1H$,I12,/' LEVELIZED CARRYING CHARGES',T28,I13,' $',T51
    &       ,2PF6.1,1H%,T62,1H$,I12,//' LEVELIZED ANNUAL REQUIREMENTS',
```

```
     &         T62,1H$,I12)
 1260 FORMAT(///' FIRST YEAR BUSBAR COST OF POWER',T55,F14.2,
     &           ' MILLS/KWH',/' LEVELIZED ANNUAL BUSBAR COST OF POWER',
     &          T55,F14.2,' MILLS/KWH')
 1270 FORMAT(///' PARTICULATE COST EFFECTIVENESS',T45,F14.2,'  $/TON',
     &          /'          SO2 COST EFFECTIVENESS',T45,F14.2,'  $/TON',
     &          /'          NOX COST EFFECTIVENESS',T45,F14.2,'  $/TON')
      END
```

```
$DEBUG
C
      FUNCTION RAT(ITYPE,IM,IY)
C
C       THIS SUBROUTINE CALCULATES COST ADJUSTMENT FACTORS FOR ANY OF THE
C     FOUR COST CATEGORIES USED IN IAPCS.  THESE ARE:
C
C         1. CE PLANT COSTS (USED FOR ALL MODULES EXCEPT LFGD)
C         2. MATERIAL COSTS (CE EQUIPMENT, MACHINERY AND SUPPORTS)
C         3. INSTALLATION LABOR COSTS (CE CONSTRUCTION LABOR)
C         4. O&M COSTS (PEI DEFINED; INDEX OF 100 IS 1982)
C
C       ADJUSTMENT FACTORS ARE CALCULATED USING ONE OF THE FOLLOWING TWO
C     METHODS:
C
C         FOR COSTS BETWEEN THE BASE DATE (JUNE, 1982) AND THE DATE OF
C         THE LATEST AVAILABLE COST INDICES (USER DEFINED IN PARMFILE),
C         THE COST ADJUSTMENT FACTOR IS CALCULATED BY LINEAR INTERPOLATION
C         BETWEEN THE BASE YEAR INDEX AND THE LATEST AVAILABLE INDEX.
C
C         FOR COSTS BEYOND THE LATEST AVAILABLE INDEX, COSTS ARE ESCALATED
C         WITH THE ANNUAL INFLATION RATE FROM THE DATE OF THE LATEST AVAIL-
C         ABLE COST INDICES.
C
C         IT IS NOT POSSIBLE TO ADJUST COSTS TO DATES BEFOR JUNE, 1982.
C
C
C       INPUTS:
C         IY - YEAR TO WHICH THE RETURNED FACTOR WILL ADJUST THE COST
C              FROM 1982 DOLLARS
C
C         IM - MONTH OF COST IN THE ABOVE YEAR
C
C         ITYPE - TYPE OF COST TO BE ADJUSTED (SEE 1-4 ABOVE)
C
C         CEYR - DATE OF LATEST AVAILABLE COST INDICES
C
C         CE(1) - LATEST AVAILABLE CE PLANT INDEX
C
C         CE(2) - LATEST AVAILABLE CE EQUIPMENT INDEX
C
C         CE(3) - LATEST AVAILABLE CE LABOR INDEX
C
C         CE(4) - LATEST AVAILABLE O&M INDEX
C
C         INF - ANNUAL INFLATION RATE
C
C       OUTPUT:
C         RAT - FUNCTION RETURN VALUE; COST ADJUSTMENT FACTOR
C
C       FILES:
C         NONE
```

```
C
C       CALLS:
C         NONE
C
C       CALLED FROM:
C         RATMOD - PARM FILE COST FACTOR ADUSTMENT ROUTINE
C
C         DRIVER - MAIN LINE OF MODULES; ADJUSTS COSTS TO OUTPUT YEAR
C
      REAL INF
      DIMENSION CE(4),CEB(4)
      COMMON /PARM/ P(500)
      EQUIVALENCE (CE(1),P(402))
      DATA BASE,CEB/82.5,314.0,336.2,263.9,100.0/
C
C     CONVERT DATES (YYMM) TO DECIMAL NUMBERS
C
      DATECE= P(290)/100.+MOD(INT(P(290)),100)/12.
      DATE= IY-1900.+IM/12.
      IF(DATE .LT. BASE .OR. DATECE .LT. BASE) THEN
        WRITE(6,*) ' DATE ENTERED IS BEFORE JUNE, 1982. 6/82 USED.'
        IF(DATE .LT. BASE) THEN
          IY= 1982
          IM= 6
          DATE= BASE
        ENDIF
        IF(DATECE .LT. BASE) THEN
          DATECE= BASE
        ENDIF
      ENDIF
C
      INF= P(174)*.01
C
C     DETERMINE WHICH ADJUSTMENT METHOD TO USE
C
      IF(DATE .LE. DATECE) THEN
C
C       USE COST INDICES
C
        CERNGE= DATECE-BASE
        RANGE= DATE-BASE
        IF(CERNGE .NE. 0.) THEN
          XM= RANGE/CERNGE
        ELSE
          XM= 1.
        ENDIF
       WRITE(8,*) 'ITYPE,CE(ITYPE),DATE,DATECE,XF,INF,RAT,CEB(ITYPE)',
     &            ITYPE,CE(ITYPE),DATE,DATECE,XF,INF,RAT,CEB(ITYPE)
        RAT= XM*(CE(ITYPE)/CEB(ITYPE)-1.)+1.
C
C       USE ANNUAL ESCALATION RATE
C
```

```
      ELSE
        RAT= CE(ITYPE)/CEB(ITYPE)*(1.+INF)**(DATE-DATECE)
C
      ENDIF
      WRITE(8,*) 'ITYPE,CE(ITYPE),DATE,DATECE,XF,INF,RAT,CEB(ITYPE)',
     &           ITYPE,CE(ITYPE),DATE,DATECE,XF,INF,RAT,CEB(ITYPE)
      RETURN
      END
```

```
$DEBUG
C
C       THIS PROGRAM CALCULATES THE CHARGE ON CAPITAL AND O&M LEVEL-
C       IZATION FACTORS BASED ON THE EPRI TAG DOCUMENT DATED 5/82.
C       CODE TRANSLATED FROM BASIC CODE IN TAG (PP. B-3-A-20,22).
C             WRITTEN BY PEI ASSOCIATES, INC (5/85 JP)
C                                            (REV. PER EPRI ECONCC
C                                       (6/21/85) VERSION 3 4/86 JP)
C
      SUBROUTINE EPECON(CCH,OMLF,DURAT,AF)
      IMPLICIT REAL(A-H,O-Z)
      REAL ITC,K
      REAL*4 CCH,OMLF,P,AF,DURAT
      INTEGER P1,B1,TDM
      COMMON /PARM/ P(500)
      DIMENSION TEMP(15),DEPR(60)
      DATA TEMP/5.,10., 9., 8., 7.,7.,6.,6.,6.,6.,6.,6.,6.,6./
C
C*******************************************************************************
C
C     SECTION 1: CAPITAL CHARGE LEVELIZATION FACTOR
C
C*******************************************************************************
C
C     SET INPUT VARIABLES TO PARAMETERS
      ITC= P(165)*.01
      B1= P(166)
      P1= P(167)
      IF(P1 .GT. 15) P1=15
      CD= P(168)*.01
      DR= P(169)*.01
      CP= P(170)*.01
      PR= P(171)*.01
      CE= P(172)*.01
      TX= P(173)*.01
      EI= P(174)*.01
      PTI= P(175)*.01
      AER= P(176)*.01
      TDM= P(177)
C
C     CALCULATE STRAIGHT LINE DEPRECIATION
C
      SLTXDP=1./B1
C
      DO 10 I=1,60
   10 DEPR(I)=0.0
C
C     CALCULATE DEPRECIATION FOR TAX PURPOSES
C
      IF(TDM .EQ. 2) P1=B1
      DO 20 I=1,P1
C>>>    ACCELERATED DEPRECIATION BASED ON ACRS CLASS
```

```
          IF (TDM .EQ. 1) THEN
            IF(I .LT. B1) THEN
              DEPR(I)= TEMP(I)*.01
            ELSE
              DEPR(B1)= DEPR(B1)+TEMP(I)*.01
            ENDIF
C>>>      STRAIGHT-LINE DEPRECIATION OVER BOOK LIFE
          ELSEIF(TDM .EQ. 2) THEN
            DEPR(I)= SLTXDP
C>>>      STRAIGHT-LINE DEPRECIATION BASED ON ACRS CLASS
          ELSE
            IF(I .LT. B1) THEN
              DEPR(I)= 1./P1
            ELSE
              DEPR(B1)= DEPR(B1) + 1./P1
            ENDIF
          ENDIF
   20   CONTINUE
C
C       COMPUTE RETURN RATES USED TO DETERMINE CARRYING CHARGES
C
C>>>    SET DISCOUNT RATE
        IF (P(180) .NE. 0) THEN
           EFI= P(180)
        ELSE
           EFI= (CD*DR+CP*PR+CE*(1.-DR-PR))*100.
        ENDIF
        ER= 1.-DR
        EC=(CP*PR+CE*(1.-DR-PR))/ER
        BD= (1.-ITC)/B1
        SB= 1.-ITC
C
C       CALCULATE CARRYING CHARGES AND OTHER TERMS
C
        CUMDEP= 0.
        PV= 0.
        SP= 0.
        PG= 0.
        DO 30 I= 1,B1
        V= (1.+EFI*.01)**(-I)
        A= (1.-V)/(EFI*.01)
        TD= DEPR(I)
        CUMDEP= CUMDEP+TD
        RE= SB*ER*EC
        RD=SB*DR*CD
        DT=(TD-SLTXDP)*TX
        TP= TX/(1.-TX)*(BD-TD+DT+RE)
        PV= PV+TD*V
        CC= BD+DT+RD+RE+TP+PTI
        SP= SP+CC*V
        SB= SB-BD-DT
   30   CONTINUE
```

```
C
      CCH= SP/A
C
C*****************************************************************************
C
C     SECTION 2: O&M LEVELIZING FACTOR
C
C*****************************************************************************
C
      EA= (1.+AER)*(1.+EI)-1.
      CM= EFI*.01
      AN= ((1.+CM)**B1-1.)/(CM*(1.+CM)**B1)
      K= (1.+EA)/(1.+CM)
C
      WRITE(8,*) 'EA,CM,AN,B1,K,DURAT',EA,CM,AN,B1,K,DURAT
      OMLF= K*(1.-K**B1)/(AN*(1.-K))
C
C*****************************************************************************
C
C     SECTION 3: ALLOWANCE FOR FUNDS USED DURING CONSTRUCTION
C
C*****************************************************************************
C
      Z= (1.+EFI*.01)/(1.+EI)
      IF(DURAT .LE. 0.) THEN
        AF= 1.0
      ELSE
        AF= (1.-Z**DURAT)/((1.-Z)*DURAT)
      ENDIF
      RETURN
      END
```

```
$notstrict
C
      SUBROUTINE PARMB(PARAM)
C
C   PARAMETER EDITOR II
C      FOR IAPCS2
C
C
C      WRITTEN BY    JEFF WINGET
C                    PEI ASSOCIATES, INC.
C
C      3-14-85
C
      LOGICAL CHANGE(500)
      COMMON /PARM/P(500)
      COMMON /PEDIT/ CHANGE,FILE1
      CHARACTER*4 EXT,TEMP1,TEMP2
      CHARACTER*1 PARAM(8)
      CHARACTER*12 FILE1,FILECH
      CHARACTER*8 FILEN
      INTEGER Y,Z,Z1,FORM,IAPCS(9),FMB,CT,BOTTOM,TYPCT,CLNLVL,YY(500)
      LOGICAL EX,TVA,KO(2)
      COMMON /INPUT/    BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,
     &          PNAROM,ALKROM,CLROM,FEOROM,PMROM,VROM,FCROM,HHVCLN,PSCLN,
     &                PACLM,CSTCLN,PNACLN,ALKCLN,CLCLN,FEOCLN,PMCLN,
     &          VCLN,FCCLN,FMB,NR,IS,CT,TYPCT,CLNLVL,BOTTOM,NM,IAPCS,KO,
     &                TVA
      DATA TEMP1/'.TVA'/
      DATA TEMP2/'.EPR'/
C
 1000 FORMAT (3X,F10.0)
      DO 7 I=1,500
         CHANGE(I)=.FALSE.
    7 CONTINUE
      IF (TVA.EQV..TRUE.) EXT=TEMP1
      IF (TVA.EQV..FALSE.) EXT=TEMP2
      CALL FTRIM(PARAM,FILECH,EXT,FILEN,EC)
      FILE1=FILECH
      OPEN (1,FILE=FILE1,STATUS='OLD')
      REWIND (1)
      READ (1,1) NP
    1 FORMAT (I4)
      DO 100 I=1,NP
         READ(1,1000) P(I)
  100 CONTINUE
      RETURN
      END
```

```
C
      SUBROUTINE HELP(HFILE)
C
C
C     THIS SUBROUTINE DISPLAYS AN ASCII HELP FILE BY READING IN ONE
C     SCREEN OF INFORMATION AT A TIME, AND THEN WRITING IT OUT.
C     IT REQURES PEI'S ANSI.SYS SCREEN UTILITIES.  IT USES UNIT
C     30 FOR THE HELP FILE.
C
      CHARACTER*80 DUMMY
      CHARACTER*12 HFILE,D*1
      CHARACTER*79 REC(24)
      DATA DUMMY/'
     &                    '/
      OPEN(30,FILE=HFILE,STATUS='OLD')
   10 CALL CLS
      DO 20 I=1,21
   20    READ(30,1000,END=99) REC(I)
      DO 30 I=1,21
   30    WRITE(6,2000) REC(I)
      CALL LOCATE(1,23,DUMMY)
      CALL SCRMOD(1)
      WRITE(6,3000)
      READ(5,'(A1)') D
      CALL SCRMOD(0)
      GO TO 10
   99 IF(I .EQ. 1) RETURN
      J= I-1
      DO 100 I= 1,J
  100   WRITE(6,2000) REC(I)
      CALL LOCATE(1,23,DUMMY)
      CALL SCRMOD(1)
      WRITE(6,3000)
      READ(5,'(A1)') D
      CALL SCRMOD(0)
      RETURN
 1000 FORMAT(A)
 2000 FORMAT(1X,A)
 3000 FORMAT(28X,'HIT [RETURN] TO CONTINUE',28(1H ))
      END
```

```
$debug
      SUBROUTINE RATMOD(IYOC,IMOC,PRAT)
      CHARACTER*4 BUFF,HT*1
      COMMON /PARM/P(500)
      DIMENSION PRAT(4)
   91 WRITE (6,1000)
1000  FORMAT (' Input reference year of costs <YYMM>, '
     & 'or ''H'' for Help '\)
      READ (5,'(A)',ERR=91) BUFF
      READ (BUFF,'(A1)',ERR=91) HT
      IF(HT .EQ. 'h' .OR. HT .EQ. 'H') THEN
         CALL HELP('COSTHELP.DOC')
         CALL CLS
         GO TO 91
      ELSE
         READ(BUFF,'(2I2)',ERR=91) IYOC,IMOC
         IF(IYOC .LT. 82 .OR. IMOC .LT. 1 .OR. IMOC .GT. 12) GO TO 91
      ENDIF
      IYOC= IYOC+1900
      DO 500 I=1,4
        PRAT(I)=RAT(I,IMOC,IYOC)
  500 CONTINUE
      RETURN
      END
```

```
$LARGE
$NOFLOATCALLS
      SUBROUTINE IAPPAS(DIREXC)
C
      DIMENSION DIREXC(9)
      COMMON /OPTIM/BYPSO2,BYPMST,BYPCFM
      COMMON /OPT/     BYPFGD,BYPFF,EFFESP,EFFLSD,EFFLIM
      COMMON/BIG1/BAGDLP,BAGRAT,BAGCOST,BGLIFE,PSSO2X,EBYPAS,RS,PSC,
     1 CCSA,EFFPS,ESPDLP,RESIST,SCARAT,FANHP,FANLC,FANMC,DELTAP,FILTSA,
     2 PSF,FILRAT,H,TL,VD,VE,TGASID,SKLAND,DRAINM,DRAINL,CLAYSK,WPSO3,
     3 WPSO4,WPCO3,WPINS,ASHCLN,WINDEX,PDMAX,FLOOD,SRAIR,OXAIR,OXSCFM,
     4 PRES,SO2RT,SO2RV,SO2RST,SDFEE,VPD,VT,HVCLN,HPTONW,
     5 ISLUDG,IFIXS,JSR,IVPD,IT,LON,LGMAX,ISTAND,NSBR,
     6 KEPASS,KPASO2,ICLAR,JSSO,JDIS,JSTR,JGPM,IRH,NSBF
      COMMON /COSTS/ CAPCST(10,8),ANN(25),INDCST(10)
      COMMON /STREAM/ PPH,ACFM1,TEMP,PPHMST,PPHALK,PPHPRT,PPHSO2,PPHNOX
      COMMON /INPUT/BSIZE,CF,AHFX,AHXT,HHVROM,PSROM,PAROM,CSTROM,PNAROM,
     &ALKROM,CLROM,FEOROM,PMROM,HHVCLN,PSCLN,PACLN,CSTCLN,
     &PNACLN,ALKCLN,CLCLN,FEOCLN,PMCLN,FMB,NR,IS,CT,TYPCT,
     &CLNLVL,BOTTOM,NM,IAPCS(9),KO(2),TVA
      INTEGER FMB,CT,BOTTOM,TYPCT,CLNLVL,AHE,IYC
      REAL INDCST
      COMMON /PARM/ P(500)
      COMMON /MATBAL/ TABLE(8,8)
      COMMON/WGAS/WACFM,WSCFM,TWG,ASHOUT,WASHOT,WMPCT(6),WLBMH(6),WLBH
      COMMON/SKGAS/SSCFM,SACFM,TSK,SDUST,SWDUST,SMPCT,SLBMH,SLBH,TGAS
      REAL INC(10),MW,LSADD,LIMEAD,LBH(6),TOTFLO(20),WLBH(6)
      REAL *8 PCTMNT,PDMNTP,XINFLA,PCTOVR,PCTADM,UNDCAP,PCTINS,UC(4
     1        0)
      REAL *8 DSBCAP,DALAND,DWRCAP,DTOFLO,MGWATT,RMGWAT,KBTU,DLBH3
      REAL *8 DWLBH3,DWPS,AQ(40),DDIREC,DPOND,DSDFEE
      REAL *8 RMAQ(40),AQLAB,UTAQ(40),AQANAL,RMC(40),UCLAB,UTC(40)
      REAL *8 UCANAL,AQLLF,UCLLF,IU,IV,KCONVR
      LOGICAL TVA,KO
      REAL M(10)
      INTEGER*4  BTUCOL,YRREV
      CHARACTER*4 IEND,INEXT
      DIMENSION NOYR(3),INVEST(3),IU(50),IV(50),UNRERN(29,1),
     3 IB1(50),IB2(50),ITEMP(25),ITAC(34),IC(50),SAVUC(40),INVSTS(3),
     5 IUNDPR(50),UNR(25),AX(25),NSAV(50),ISAV(50),JSAV(50),SAV(50)
      DIMENSION AQSHIP(11),KDEPR(50),KCONVR(11),UNRRN(11,121),PCTIF(29),
     3 SAVUC2(11),UNRR(11),ISHIP(12),AXSHP(11),PCTIFS(11),IFEERE(50),
     5 UCSHIP(11),ITAREV(50),IFEER2(50)
      DATA IEND,INEXT/'END ','NEXT'/
      DATA NUMRAW,NUMUTL,NUMUNT/12,8,23/
      DATA M/44.01,36.46,64.06,32.,28.02,18.016,12.01,35.45,32.06,1.008/
      WRITE(8,*) ' IAPPAS INDCST=',INDCST,TVA
C*****************************************************************************
C
C     MATERIAL BALANCE CALCULATIONS
C
C>> ACFM: NAME CHANGED TO PROTECT TVA 'ACFM'
```

```
      ACFM1=WACFM*(TWG+460)/(TSK+460)*(14.696/PRES)
C
C     MOISTURE (LB/HR)
C
      PPHMST=WLBMH(6)*M(6)+BYPMST*M(6)
C
C     SO2 (LB/HR)
C
      PPHSO2=WLBMH(3)*M(3)+BYPSO2*M(3)
C
C     FLUE GAS MASS FLOW RATE (LB/HR)
C
      PPH= ( WLBMH(1)*M(1)/(1.-BYPFGD))+(WLBMH(2)*M(2)/(1.-BYPFGD))+
     & (WLBMH(4)*M(4)/(1.-BYPFGD))+(WLBMH(5)*M(5)/(1.-BYPFGD))+PPHMST+
     & PPHSO2
C
C     TEMPERATURE (DEG. F)
C
      TEMP1=TEMP
      TEMP=(1.-BYPFGD)*(TSK+460)+BYPFGD*(TEMP1+460)-460
C
C     FLUE GAS VOLUMETRIC FLOW RATE ADJUSTED FOR TEMPERATURE (ACFM)
C
      ACFM1=ACFM1*(TSK+460)/(TEMP+460)+BYPCFM*(TEMP1+460)/(TEMP+460)
C*****************************************************************************
C
C     DIRECT CAPITAL COST CALCULATIONS
C
C     CAPCST(8,
C               2 -- MATERIAL HANDLING
C               3 -- FEED PREP
C               4 -- GAS HANDLING
C               5 -- SO2 SCRUBBING
C               6 -- OXIDATION
C               7 -- REHEAT
C               8 -- SOLID SEPARATION
C
      DO 11000 I=1,7
         CAPCST(8,I+1)=DIREXC(I)*1000*P(298)
11000 CONTINUE
C>>>  TOTAL DIRECT CAPITAL COST
      CAPCST(8,1)=1000*DIREXC(8)*P(298)
C*****************************************************************************
C
C     ANNUAL COST CALCULATION: ONLY WORKING CAPITAL IS CALCULATED
C     HERE, OTHERS ARE IN WORKCP OR WASTE.
C
      ANN(10)=ANN(10)+CAPCST(8,1)*P(366)*.01
C*****************************************************************************
C
C     INDIRECT CAPITAL COST CALCULATIONS
C
```

```
C
      IF (TVA) THEN
       P(358) = INT((P(358)-1.) + 2.*(BSIZE -100.)/900. + .5)
       P(359) = INT((P(359)-1.) + 2.*(BSIZE -100.)/900. + .5)
       P(360) = INT((P(360)-1.) + 3.*(BSIZE -100.)/900. + .5)
       P(361) = INT((P(361)-1.) + 2.*(BSIZE -100.)/900. + .5)
      ENDIF
      WRITE(8,*) ' INDCST',INDCST
C
C      CALCULATE ENGINEERING DESIGN AND SUPERVISION (TVA) OR
C      GENERAL FACILITIES (EPRI)
C
      INC(1) = P(358)/100. * CAPCST(8,1)
      INDCST(1) = INDCST(1) + INC(1)
C
C      CALCULATE A-E CONTRACTOR(TVA) OR ENG. AND HOME OFF. FEES (EPRI)
C
      INC(2) = P(359)/100. * CAPCST(8,1)
      INDCST(2) = INDCST(2) + INC(2)
C
C      CALCULATE CONSTRUCTION EXPENSE (TVA) PROJECT CONTINGENCY (EPRI)
C
      INC(3) = P(360)/100. * CAPCST(8,1)
      INDCST(3) = INDCST(3) + INC(3)
C
C      CALCULATE CONTRACTOR FEES (TVA) OR PROCESS CONTINGENCY (EPRI)
C
      INC(4) = P(361)/100. * CAPCST(8,1)
      INDCST(4) = INDCST(4) + INC(4)
C
C      SUM DIRECT AND INDIRECT CAPITAL COSTS TO HAVE A BASE FOR THE
C      REMAINDER OF THE INDIRECT COSTS.
C
      COST = CAPCST(8,1) + INC(1) + INC(2) + INC(3) + INC(4)
C
C      CALCULATE CONTINGENCY (TVA) OR SALES TAX (EPRI)
C
      IF (TVA) THEN
            INC(5) = P(362)/100. * COST
         ELSE
            INC(5) = P(211)/100. * COST
         ENDIF
      WRITE(8,*) ' P(211),INC',P(211),INC
      INDCST(5) = INDCST(5) + INC(5)
C
C      INCLUDE IC(5) IN RUNNING COST BASE TOTAL
C
      COST = COST + INC(5)
C
C      CALCULATE ROYALTIES (TVA) OR ROYALTY ALLOWANCE (EPRI)
C
      IF (TVA) THEN
```

```
          INC(6)=P(377)/100. * COST
        ELSE
          INC(6) = P(364)/100. * COST
      ENDIF
      INDCST(6) = INDCST(6) + INC(6)
C
C      CALCULATE INTEREST DURING CONSTRUCTION (TVA)
C
      IF (TVA) THEN
        INC(8) = P(364)/100. * COST
        INDCST(8) = INDCST(8) + INC(8)
      ENDIF
C
C       CALCULATE ALLOWANCE FOR STARTUP AND MODIFINATIONS (TVA) OR
C                 INITIAL CATALYST (EPRI)
C
      IF(TVA) THEN
          INC(9)= P(363)/100.*COST
      ELSE
          INC(9)= P(222)/100.*COST
      ENDIF
      INDCST(9) = INDCST(9) + INC(9)
C
C       CALCULATE LAND (TVA AND EPRI)
C
      INC(10) = P(89)
      INDCST(10) = INDCST(10) + INC(10)
C
      WRITE(8,*) ' AFTER IAPPAS: INDCST',INDCST
      RETURN
      END
```